DOCUMENT RESUME

ED 034 395 EM 007 514

AUTHOR	Blum, Ronald, Ed.
TITLE	Computer-Based Physics: An Anthology.
INSTITUTION	Commission on Coll. Physics, College Park, Md.
PUB DATE	Sep 69
NOTE	205p.
EDRS PRICE	EDRS Price MF-$1.00 HC-$10.35
DESCRIPTORS	College Instruction, *Computer Assisted Instruction, Curriculum Development, Data Analysis, Input Output Devices, *Interaction, Motion, Physical Environment, Physics Curriculum, *Physics Instruction, Probability, *Problem Solving, Programing, Simulated Environment, *Simulation, Spectrograms

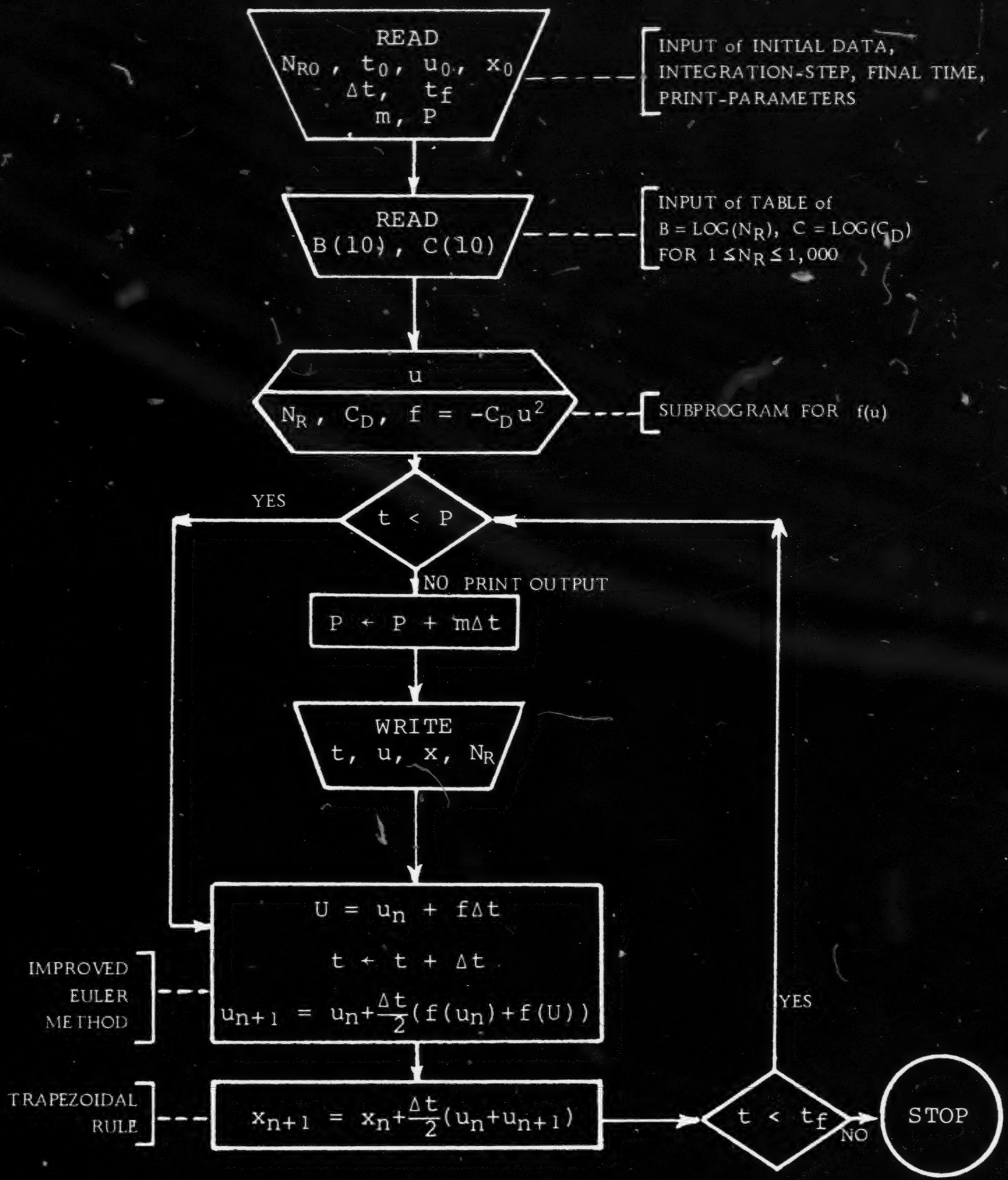
ABSTRACT

Designed to serve as a guide for integrating interactive problem-solving or simulating computers into a college-level physics course, this anthology contains nine articles each of which includes an introduction, a student manual, and a teacher's guide. Among areas covered in the articles are the computerized reduction of data to a Gaussian distribution, "free-style" input which frees the student from format restrictions, the effects of integrating physics concepts with computer programing, harmonic motion, relativistic and nonrelativistic two-body collisions, simulation of relativistic collisions in a bubble chamber, further computer simulations involving mass spectrometers and physical systems (including the roulette wheel or Monte Carlo approach), vacuum and low-velocity ballistics in a conversational approach, and computerized curriculum open-ending to provide increased realism and relevance. The student manual and teacher's guide sections clarify and expand the information presented in each introduction. Computer listings, subroutine, and structure maps are provided also for many of the articles. (Author/SP)

ED034395

COMPUTER-BASED PHYSICS
An Anthology

COMMISSION ON COLLEGE PHYSICS


READ
NRO , t0 , u0 , x0
Δt, tf
m, P
INPUT of INITIAL DATA, INTEGRATION-STEP, FINAL TIME, PRINT-PARAMETERS
READ
B(10), C(10)
INPUT of TABLE of B = LOG(NR), C = LOG(CD) FOR 1 ≤ NR ≤ 1,000
u
NR , CD , f = -CD u²
SUBPROGRAM FOR f(u)
YES
t < P
NO PRINT OUTPUT
P ← P + mΔt
WRITE
t, u, x, NR
U = un + fΔt
t ← t + Δt
un+1 = un + Δt/2 (f(un) + f(U))
IMPROVED EULER METHOD
TRAPEZOIDAL RULE
xn+1 = xn + Δt/2 (un + un+1)
t < tf
YES
NO
STOP

EM007514

ED034395

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

COMPUTER-BASED PHYSICS: AN ANTHOLOGY

Editor
Ronald Blum

Contributors

Ronald Blum
Commission on College Physics

David T. Grimsrud
Muhlenberg College

Thomas R. Harbron
Anderson College

Jeffrey Jalbert
Denison University

John Kenyon
State University of New York at Geneseo

Kenneth F. Kinsey
State University of New York at Geneseo

R. C. Mikkelson
Macalester College

Charles W. Miller
Anderson College

Paul A. Smith
Coe College

Anton F. Vierling
United States Naval Academy

Dale Winder
Colorado State University

Published by the Commission on College Physics, 4321 Hartwick Road, College Park, Maryland 20740.

September 1969

PREFACE

The articles contained herein are the outgrowth of a computer workshop organized by Dr. John W. Robson* for the Commission on College Physics in 1968. He succeeded in forming interested individuals into a Computer Working Group which met several times during the year; their contributions constitute this volume.

Almost every contributor presented the Editor with a wealth of material and those segments were selected for inclusion which seemed to the latter's judgment to fit most harmoniously together. This was done with considerable anguish and soul-searching; one result of this synthesis is that a given contribution does not necessarily represent the author's best or most unique work, but, in most cases, only the tip of the iceberg. In consequence, the Editor has thumpingly urged the authors of the lost masterpieces to make them available to the general public through the Computer Library for Instruction in Physics (see the American Journal of Physics 35, 273 (1967)).

In using the computer in physics education, as distinct from research, we are not only interested in ways of solving problems, but in how the computer can add a new dimension to the nature and content of the curriculum through its influence on the topics selected and their mode of presentation. In order to remain relevant to future needs, basic undergraduate courses must be appropriately modified to reflect the new points of view associated with computer applications, numerical analysis must be integrated into course work, and students should be given programming instruction at an early stage in their education.

The use to which we put the computer depends on the available facilities. From a pedagogical standpoint these are of three types--interactive, semi-interactive and noninteractive. Large, expensive installations which process programs in batches are noninteractive. Smaller and cheaper machines may allow a student to receive his output within a few minutes, alter his program and resubmit it, if necessary. These are semi-interactive in that receipt of output is immediate, but errors or program alterations necessitate terminating the program. The future widespread use of the computer in the physics curriculum will result from the availability of interactive terminals, where the student can receive output immediately, correct errors and input new instructions or data without terminating his connection with the computer. Hence, the emphasis in this volume has been on interactive and semi-interactive facilities.

The computer may be used in the classroom in any of four ways:

---

*Presently, Dr. Robson is with the Department of Physics at the University of Arizona, Tucson.

as an administrator, tutor, simulator or calculator. In the first mode it may be used simply to administer and grade exams and perform other onerous clerical chores. In the second mode it may, in addition, tutor the student; i.e., correct his errors by hint, precept or example and so lead him along various paths which are determined by his previous answers and designed to aid him to overcome his deficiencies.

The computer can simulate physical reality either as a "black box" or as a loaded roulette wheel. The "black box" may be a program representing a physical system into which the student enters the values of certain physical quantities and then observes the output from the program in place of an actual experiment. On a time-shared teletype the student can easily "fiddle with the knobs," i.e., change his inputs and observe the results. In the roulette wheel, or Monte Carlo, method the computer generates random numbers which can be used to simulate phenomena in which chance in a factor. For example, one could simulate a baseball game using the batting averages of the players. A player batting .325 would be allowed a hit each time the three-digit random number representing his time at bat was between 000 and 324 and called "out" if it was between 325 and 999. Such methods can be applied to problems of gaseous diffusion, radioactive decay, scattering, etc.

However, it is as a calculator and solver of problems that the computer should have its greatest impact on physics education. And rather than have students use programs they do not understand, it would be preferable to integrate the computer into physics at an early stage. The major problem is the lack of textual materials and programs, and the need for wider dissemination of those which do exist. It is to serve this need that this volume was conceived, and for this reason, also, programs are presented in FORTRAN or BASIC, the two most popular languages in use today.

Since the individual authors generally performed their work prior to the organization of the Computer Working Group, some duplication was inevitable, and, in the case of the harmonic oscillator--probably the world's most "programmable" elementary problem--was considered beneficial. Thus, we have presented the work of Vierling which primarily illustrates the application of an advanced fourth-order Runge-Kutta method to the theoretical problem, as well as that of Grimsrud, designed for use in connection with an elementary pendulum experiment.

The volume opens with "Data Reduction" by Smith, which is in the nature of a prerequisite to computer-oriented physics, since it deals with the use of the computer to reduce data to a Gaussian distribution and also describes an auxiliary program which may be used to interpret "free-style" input, in which the students' input to the program is freed from the usual tedious and confusing restrictions on format. Following Vierling and Grimsrud is Winder's article, which describes relativistic two-body collisions, Grimsrud having already presented the nonrelativistic case. This leads, quite naturally, to Mikkelson's simulation of relativistic collisions in a bubble chamber. The next articles, by Harbron & Miller and Kinsey & Kenyon, further illustrate the use of the computer as a simulator; the latter illustrating, too, the Monte Carlo approach.

Although we have generally elected to avoid the tutorial mode in this monograph, Jalbert's treatment of vacuum and low-velocity ballistics does provide an example of a conversational approach. Finally, Blum's article attempts to demonstrate how the computer can be employed in the construction of an open-ended curriculum in the sense that without much broadening of the conceptual base, as given by Jalbert, the student can be enabled to treat problems of greatly increased realism and relevance.

Although the Editor has required contributing authors to cast their articles in the same format--Introduction, Student Manual, and Teacher's Guide--a conscious attempt has been made to preserve individual nuances of style and approach on the grounds that these subtleties are themselves of interest to the practicing pedagogue. However, the Editor takes no responsibility for the absolute rectitude of the contributing authors, with the exception of the final article in this anthology. Criticisms and suggestions are welcomed, and remarks directed to particular authors will be forwarded to them by the Editor. Individual articles may be ordered separately through the Commission on College Physics, hence the unusually verbose footnotes scattered through the text.

The Editor acknowledges the invaluable assistance of Mrs. Faye von Limbach who prepared the typescript and the flow charts and whose pithy observations often served as a useful stimulus. Equally valuable were the tireless and enthusiastic efforts of Mr. Lee A. Fowler who checked out many of the programs in this work. Miss Kathryn E. Mervine also assisted in the editing, and the project enjoyed the wholehearted support of Dr. John M. Fowler, Director of the Commission on College Physics.

Members and friends of the Computer Working Group, in addition to those mentioned above, included Alfred Bork, University of California, Irvine; David J. Cowan and Richard T. Mara, Gettysburg College; S.A. Elder, U.S. Naval Academy; Russell K. Hobbie, University of Minnesota; Arthur Luehrmann, Dartmouth College; Anatole Shapiro, Brown University; Harold Weinstock, Illinois Institute of Technology; Ronald Winters, Denison University; Claude Kacser, Leonard Rodberg and Sanders N. Wall, University of Maryland.

By and large the papers presented here do not claim to be unique or particularly efficient. Hopefully, as a collection, they are all of pedagogical interest as regards their presentation, the problem treated and the experiences of their creators. We hope it may serve as a guide and companion to those who wish to integrate computers meaningfully into the context of their course work.

Ronald Blum
Commission on College Physics

TABLE OF CONTENTS

ARTICLE PAGE

Preface iii

Data Reduction 1
Paul A. Smith

Harmonic Motion 35
Anton F. Vierling

Two Experiments 55
David T. Grimsrud

Relativistic Two-Body Collisions 73
Dale Winder

A Simulated Accelerator Laboratory 85
Ray C. Mikkelson

Computer Simulation of a Mass Spectrometer 107
Thomas R. Harbron and Charles W. Miller

Randomness and Radioactive Decay 125
Kenneth F. Kinsey and John Kenyon

Ballistics 151
Jeffrey Jalbert

Opening the Other End 169
Ronald Blum

DATA REDUCTION

Paul A. Smith

Coe College
Cedar Rapids, Iowa

INTRODUCTION

This paper describes "GAUS," a program designed to compute the mean and standard deviation of a collection of laboratory data, as well as its goodness of fit to a Gaussian (normal) distribution by means of the chi-squared test. Also described is a general purpose auxiliary program, "FREE," which allows the students to input data in a format-free mode. This greatly simplifies the use of the computer in the classroom or laboratory, since data can be written in a natural way and the students need not learn the intricate and often tedious details of formatting, a fruitful source of time-consuming errors.

Use of computer facilities to support undergraduate physics laboratory instruction started at Coe College in the academic year 1965-66. A number of programs were written to facilitate data reduction in an intermediate electricity and magnetism laboratory and later "GAUS" was developed to help students in an introductory laboratory gain insight into error of measurement. For the first two years Coe made use of the IBM 7044 located at the University of Iowa, twenty miles distant, via a courier service that provided overnight turn-around. Initially, "GAUS" was used via optical sense-mark cards developed by the Measurement Research Center in Iowa City and requiring only a number two pencil for marking.

In the spring of 1967, under an NSF grant, Coe installed an IBM 1130 with 8K core and 1/2 M disk; that summer existing programs were modified for the 1130 and, with the help of high school students, an extensive library of simple but generally useful subroutines was developed to simplify further programming. This included the first version of FREE and its associated supportive subroutines. In the fall of 1967 and again in 1968, the author taught an introductory physics laboratory with calculus, in which computer use played a major role. The programs were introduced to the students as black boxes with little or no reference to their details of operation. Emphasis was on least squares fits of experimental data for linear acceleration, damped linear acceleration, and damped simple harmonic motion. It has taken typical students three to five weeks to really grasp the concept of least squares fits in these situations. About 280 students have made use of "GAUS" and found it a stimulating educational experience.

On the basis of three years' experience, the author has the following recommendations to make:

1. Simplify input-output problems with a free style card reader.
2. Develop a balanced diet of very simple programs that students can write and/or modify, along with complicated programs such as those used here.

3. Don't expect to teach students very much FORTRAN while they are simultaneously engaged in more traditional laboratory experiments.
4. Be alert for students who have "personal relations" problems with the computer. Some may not get assigned work done because they feel an antipathy toward the computer, others may lag behind due to their infatuation with it and associated diversions. Both will need sympathetic help!

The author wishes to acknowledge the assistance of Dr. Joseph Kasper of Coe College and Dr. Ronald Blum of the Commission on College Physics, who are responsible for much of the documentation on the Gaussian distribution in the Student Manual which follows.

STUDENT MANUAL

Gaussian Distribution

Suppose that a measurement of a physical quantity is made, where it is known that completely random deviations in the measurement occur. Suppose further that the measurement is made repeatedly, with all sources of systematic error eliminated. The theory of probability demonstrates that if one plots the *frequency of occurrence* of a given value of N as a function of *the value* of N, and if one collects very large numbers of data, then the graphical representation of the results will ideally be in the form of a symmetrical curve known as a Gaussian curve, or Gaussian distribution.

A typical Gaussian distribution, symmetric about the mean value, $\bar{N}$, is shown in Figure 1; the ordinate, $P(N)$, is the *relative* probability (or "probability density") that any particular measurement, N, will occur. While in reality the quantity measured may only take on certain discrete values, we shall find that the continuous distribution is in fact a very useful tool for the reduction of such data. The analytical formula for this curve is

$$P(N) = (2\pi\sigma^2)^{-1/2} \exp[-(N - \bar{N})^2/2\sigma^2] \qquad (1)$$

where σ^2 is an independent parameter known as the *variance* of N and its root, σ, is called the *standard deviation* of N; $\exp(x) = e^x$, another notation for the exponential function. The constant multiplier $(2\pi\sigma^2)^{-1/2}$ is chosen such that the total area under the curve between the limits $-\infty \leq N \leq \infty$ is exactly unity. Thus, it must follow that the area under the relative probability curve between, say, N_1 and N_2 *is* the theoretical probability that a measurement of N will fall between those two values. Furthermore, although there is some inconsistency in special cases where N can never actually be negative, such as the count rate from a radioactive source, this discrepancy is negligible *if* the area under $P(N)$ for negative N is only of the order of a few percent, which is generally the case in our experiments.

Figure 1 shows two Gaussian curves with the same mean value $\bar{N} = 9$, but two different values of standard deviation, σ. Although both curves peak at $N = \bar{N} = 9$, the curve with greater deviation is wider and, consequently, lower, since the area under each curve must be unity. If one made but one measurement of the physical quantity involved, and knew that the lower curve applied then he would have little confidence in it, because on the grounds of probability there would be a good chance that his one value was considerably greater or smaller than the mean. However, if he made one measurement and knew that the higher curve applied, his confidence in the measurement would be much higher. This degree of confidence, or probable deviation from the mean, is stated in various ways.

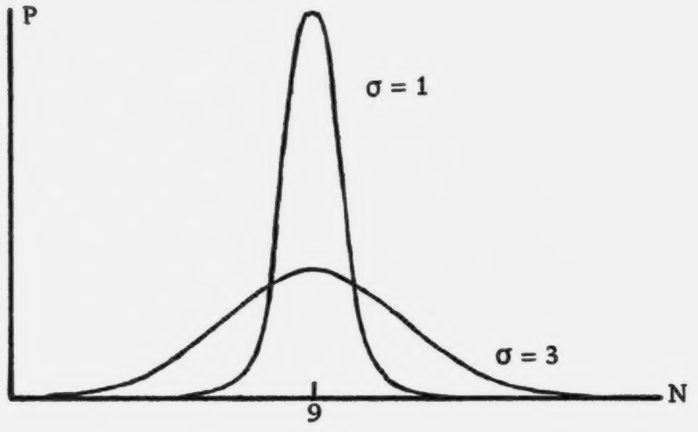


Figure 1. Gaussian distributions for $\bar{N} = 9$, $\sigma = 1$, and $\sigma = 3$.

It is a truly remarkable aspect of physical measurements involving random errors that:

a) If a measurement of some physical quantity is made, then, because of random fluctuations, it cannot be taken as entirely reliable, even if there are no instrumental or human errors present. However, the known Gaussian distribution can be used to furnish a meaningful estimate of the validity of the measurement.

b) If sets of measurements are made, then they can be compared with the ideal Gaussian curve, and deviations in the actual distribution can be used to check the validity of the results.

The standard deviation is a measure of the random variability of individual measurements and hence of the confidence we may place in them. It is often useful to think in terms of units of σ about the mean value, as shown in Figure 2. One can see that 68.3% of the time an individual measurement taken at random will lie within $\pm\ \sigma$ of the mean value, and 95.7% of the time within $\pm\ 2\sigma$ of the mean. Furthermore, in statistical theory, it is shown that the mean value, while it may not be precisely the same as the true value, is the best estimate we can form of the true value; our results are no less "scientific" for having taken account of the harsh realities of random errors. It should also be clear from Figure 2 that when $\bar{N}$ is greater than 2σ the fact that $P(N)$ is also defined for $N < 0$ will not cause any serious errors in estimating probabilities in radioactive counting experiments from the Gaussian distribution.

In general $\bar{N}$ and σ are independent of each other; while repeated measurements of the period of a pendulum yield an average value depending primarily on the physics of the pendulum, the standard deviation of the measurements depends on the instrument and technique of measurement. However, there is one important situation in which a very interesting relationship exists between $\bar{N}$ and σ. Whenever the measurement consists of counting independent events occurring at random in successive intervals of time (or space), then

$$\sigma = \sqrt{\bar{N}}$$

provided that $\bar{N}$ is moderately large (i.e., $\bar{N} > 10$).

For example, if one counted events with a Geiger tube over a radioactive source and got 100 counts in some time interval, such as ten seconds--hence, the standard deviation is $\sigma = 10$ counts--then 68.3% of the time such a single measurement would be within ± 10 counts of the average number of events due to that particular source. That is, were we to estimate that $90 \leq \bar{N} \leq 110$, we would be right 68.3% of the time, or in 683 cases out of 1,000 such measurements. The counting rate, R, would be

$$R = (100 \pm 10) \text{ counts}/(10 \text{ sec})$$

with a "confidence level" of 68.3%. Were we willing to make a looser prediction we could say $R = (100 \pm 20)$ counts/(10 sec), with a confidence level of 95.7%.

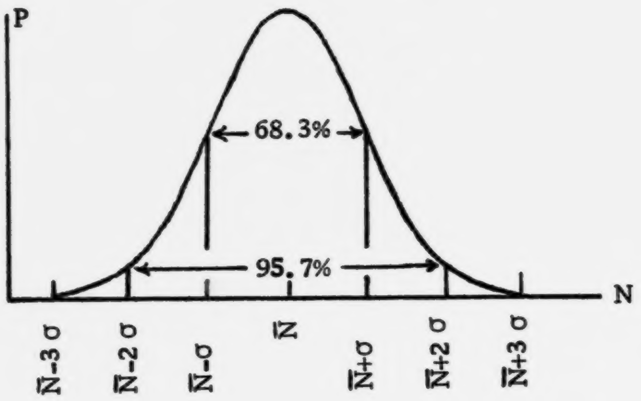


Figure 2. Gaussian distributions in intervals of σ.

The standard deviation also has the significance that if we form the quantity $(N - \bar{N})^2$ for each measurement of N and average all such values, we obtain σ^2; hence, σ may also be known as the root-mean-square (rms) error of a variable. In counting-situations, where $\sigma = \sqrt{\bar{N}}$, even one measurement of N affords a basis for estimating σ. However, in most situations one needs several measurements to form an accurate estimate of the standard deviation, according to the formula

$$\sigma^2 = \frac{\sum_{i=1}^{i=M} (N_i - \bar{N})^2}{M}$$

where a total of M measurements of counts have been made; N_i is the result of the i-th measurement. (It is actually most correct to use M-1 in the denominator of the above expression; however, this is a subtle theoretical point, and if M is large this is a negligible effect.)

Now suppose we have collected some statistical data in the form of repeated measurements. What can be made of them? They can be compared with the expected Gaussian distribution by making what is called a "chi-squared test". The quantity "chi-squared" is computed by first preparing a histogram, or bar-graph, showing the number of times (frequency) each measurement appeared, as a function of the value of that measurement. Actually, each histogram bar represents the number of times that measurements lead to values falling in a certain interval; e.g., one bar might represent how many times the count was between 90 and 100, the next bar how many times the count was between 100 and 110, etc.

When the completed histogram is at hand, it can be superimposed on the ideal or expected Gaussian distribution. There will be a difference between the actual number of values, n_j, observed to fall in the j-th interval and the expected number of values, m_j, which the Gaussian distribution predicts will fall in the interval. If, for each such interval, we compute the quantity $(n_j-m_j)^2/m_j$, the sum over all J intervals is called the chi-squared statistic.

$$\chi^2 = \sum_{j=1}^{j=J} \frac{(n_j - m_j)^2}{m_j}$$

The better the fit of distribution to data, the smaller is χ^2.

Probability theory tells us, for randomly distributed errors, the probabilities of obtaining different values of χ^2 for different values of J; thus, one can check equipment or data for systematic or nonrandom errors by comparing the fit of the data to a Gaussian curve. This information is commonly found in handbooks in the form of tables which give the probability that χ^2 will equal or exceed a certain numerical value. A sample of one line from such a table is shown below:

Number of Intervals, J	There is a probability of				
	0.99	0.90	0.50	0.10	0.01
.	that the calculated χ^2 is equal to or greater than				
.	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .
.	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .
19	7.63	11.65	18.34	27.20	36.19
.	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .
.	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .

Thus, there is no unique answer to the question: When is a fit good or bad? Instead, there is only a probabilistic answer.

For example, as the excerpt from the table shows, if one measured a count rate 19 times, plotted the distribution, plotted the expected Gaussian distribution, computed the value of χ^2, and found χ^2 to be 10, the fit would be quite good. Specifically, there would be more than a 90% chance that if the experiment were repeated the χ^2 would be greater than 10. On the other hand, if χ^2 turned out to be as high as 40, the fit would be considered very bad and it would be unlikely that the distribution represents a random selection from a set of Gaussian distributions.

In collecting data by measuring some quantity over and over, peculiar instances sometimes arise. For example, if one were consistently getting between 360 and 440 counts in one-minute intervals from a radioactive source, and then in a given one-minute interval got only 120 counts, one would be suspicious of that result. According to the Gaussian curve, such an anomalous count is perfectly possible, for the Gaussian curve runs to infinity in both directions from the mean. However, the probability of counts far from the mean drops off more and more rapidly, the farther they are from the mean. If one gets a count of 120 in a one-minute interval, after consistently getting close to 400 each minute, the thing that should really disturb us is not that this is totally impossible, but that it is highly unlikely. In fact, it may seem so unlikely that one would not want to include the result at all, because it is not typical and would throw the mean value off with more weight than it deserves. In short, one is tempted to reject that far-off value.

However, such subjective selectivity constitutes rather wanton tampering with the scientific data. The prime requisite for scientific honesty and objectivity is to let nature speak for herself, rather than to interpose the subjective bias of the experimenter. The solution to this problem is to adopt a specific criterion, expressly stated, for the acceptability of data. One such is "Chauvenet's Criterion" which states that an observation should be discarded if the probability of its occurrence in the set of observations is equal to or less than $1/(2K)$, where K is the number of observations. The table below, assuming a Gaussian distribution, gives the maximum acceptable departure of any one reading from the mean in units of σ:

Number of Observations	Maximum Departure Acceptable in Units of Standard Deviation
5	1.65
10	1.96
20	2.24
50	2.58
100	2.81
200	3.02
500	3.29
1000	3.48

Interested students may find it instructive to investigate the Poisson distribution, which, when its mean is large, has the shape of the Gaussian distribution. The ideal Poisson distribution has the peculiar property that, regardless of the magnitude of the mean, the standard deviation is equal to the square root of the mean of the distribution of values.

Instructions for Using the Computer Program "GAUS"

(1) Make repeated measurements of some physical quantity according to whatever instructions you have been given for the particular laboratory session. Take care to record the values obtained in the order you obtain them.

(2) Punch into standard IBM data cards the values you obtained:
- a. Separate different values by one or more blank positions on the card.
- b. Punch decimal points only for values with a decimal fraction part.
- c. Punch as many values on one card as you can, then continue on another card.
- d. Do not split a given value between two cards.
- e. Punch errors may be "erased" by overpunching each column of the given value with an X, if column one is blank.
- f. After your last value, enter "9999" in your data card, no decimal point!

(3) Make up a data deck as follows using pre-punched cards if they are supplied; otherwise, punch your own additional cards as needed. The symbol "b" does not stand for the letter "B" punched on a data card; it stands for a blank column with no hole punched in it. Blank columns so designated must be present. The initial "//" must be in columns one and two of the cards.

Here is how your deck of cards should look (top line represents top card):

```
//bJOBbT
//bXEQbGAUS
A card with your name or identifying number punched.
Your data cards--in order--as prepared under (2);
   be sure you enter 9999 after your last value.
//b*bJOBbEND
```

(4) Submit your program; or run it yourself, if so instructed.

Output from GAUS

The first printed line is your identification card.

Values rejected according to the Chauvenet Criterion are noted next; one or more values may be rejected at a time. After each such occasion of rejection the remaining list is again checked to see if any values in it should be rejected according to Chauvenet's Criterion. Any value that is exactly zero is automatically rejected, as it is likely to be due to a keypunch or card reading error. You may enter values arbitrarily close to zero, however; e.g., 0.0001 is permitted.

Next, the values remaining are printed out in the original order on the left, along with three additional columns: the "Running Average," "Running Sigma," and "Running Sigma of Ave." The "Running Average" is the average of the value to its left plus all preceding values in the first column; the "Running Sigma" is the standard deviation of those values; "Running Sigma of Ave." is an estimate of how confident you can be in the value of the "Running Average" value just to its left. It is, in effect, a prediction of the standard deviation you might expect to get upon making a list of average values obtained in a manner exactly as the one to its left was obtained. That is, if the average value, $\bar{x}$, of a collection of data, x, (statistics) is itself considered as a new statistic, the theory of probability predicts that a collection of average values derived from data samples of M different measurements will have a standard deviation σ_x related to the deviation $\sigma_{\bar{x}}$ of a single measurement by

$$\sigma_{\bar{x}} = \sigma_x / \sqrt{M}$$

The last values in these "Running" lists apply to the full list given, except for those values rejected.

The full list is next divided up into successive subsets to illustrate the way in which the average and standard deviation of successive subsets fluctuates about the average and standard deviation applicable to the full list. Compare these values to the last entries under "Running Average" and "Running Sigma".

The list of subset averages is now considered as a new list of values for which the average and standard deviation is computed. The same is done for the list of the standard deviations of the subsets, which may, in turn, be considered a statistic and assigned an average, $\bar{\sigma}$, and a standard deviation, $\sigma_{\sigma_{\bar{x}}}$.

The first value under "Their Sigma," which represents the standard deviation of the list of averages of subsets, should be compared with the M-th entry under "Running Sigma of Ave." above, where M = the number of values in each subset of values. These two values should be within 10 to 20 percent of each other if your list of values has a true Gaussian shape.

Your list of values is again printed out, but this time ordered according to value; also listed is the departure of each value from the mean value, i.e., the last value in the list "Running Average."

Next, a histogram is printed out along with an indication of the interval of values included in each histogram bar. This is followed by a numerical comparison of the observed histogram with the expected. "CHI2" heads the list of contributions to the chi-squared sum. "Lower" and "Upper" are bounds on the histogram bars in terms of departures from the mean. If the data is distributed according to a Gaussian distribution, then theory tells us that the value of χ^2 for data selected at random will be less than the number of histogram bars at least 50% of the time, the exact figure depending upon the exact number of bars. Thus, as a rule-of-thumb, if χ^2 is significantly larger than the number of histogram bars, we would have reason to question the validity of the Gaussian distribution as applied to that particular set of data. (Incidentally, regrouping the data to make more or less bars also affects the value of χ^2 accordingly--you can't beat the system!)

TEACHER'S GUIDE

Approximately 280 students have used GAUS over the past three and a half years. These have been students of high school age in summer NSF science programs as well as students in our introductory physics courses. The student response has been generally quite good, with a significant number of students showing genuine enthusiasm.

A variety of measurement procedures have been used to generate data for GAUS:

1. Repeated measurements of the period of a simple torsional pendulum made by hanging a rod on magnetic tape.
2. Repeated measurements of transit time for a car on an air track.
3. Radioactive decay counts--each student instructed to set his counting equipment so as to get some assigned average count rate, at least approximately, so the class as a whole can check the prediction that standard deviations will go as the square root of the mean values.

Class and/or laboratory discussion is directed to the usefulness of the standard deviation associated with measurement. In particular this is related to the question of whether certain functional relations describe the relationship between experimental values to within a reasonable degree of accuracy. The utility of the Gaussian distribution needs to be emphasized and illustrated by actually making use of it in laboratory work.

In selecting an experimental procedure to generate data for GAUS, the following considerations are important:

1. It should be possible to obtain about 100 independent measurements/hour; GAUS can handle 200 readings in its present form.
2. Repeated measurements should yield a significant scatter of values over a continuous range--specific values should rarely repeat. The histogram generated may look very strange if this condition is not met.
3. Except as you desire it, there should be no trend for the repeated measurements to lead to values tending to steadily increase or decrease. Note:

extended use of a stop watch can lead to fatigue with increased response time and variance--this might be interesting to study.

4. Avoid a change in "observer" in the middle of a set of measurements, except as you wish students to study possible effects of such changes, if any.

It has been possible at Coe College to have students go directly from the laboratory to keypunches and then to the IBM 1130 where they could run their data "open-shop". While this does help develop interest, there was also good student response even when we had delays due to courier service to and from Iowa City. With the IBM 1130 we store the program in "Core Image Format," so that execution starts within about five seconds and runs one to three minutes for typical sets of data used. Students are given the option of having their decks run for them closed-shop style if time or inclination rules against the "hands-on" operation.

The following pages contain what it is hoped are self-explanatory listings of GAUS and its associated subroutines, followed by an Appendix explaining the use of the subroutine for free-style input.

```
// *   MODIF68  05
// DUP
*DELETE             GAUS
// FOR
*IOCS(CARD,1132 PRINTER)
*ONE WORD INTEGERS
C     GAUS USES SUBROUTINES SUBSE, ORDE, COUNT, GEX, CHI, AND FREE
C     GAUS   COORDINATES THE USE OF SUBROUTINES ORDE,  COUNT, GEX, AND
C     CHI TO MAKE AN ANALYSIS OF AN UNORDERED SET OF VALUES HAVING,
C     PRESUMABLY, A GAUSSIAN DISTRIBUTION.  IT ACCEPTS THE  K  VALUES
C     IN THE VECTOR X(I) AND GENERATES THE FOLLOWING
C     AVE =  THE AVERAGE VALUE OF SET OF READINGS
C     S = THE STANDARD DEVIATION OF THE SET OF READINGS
C     INTV= THE NUMBER OF INTERVALS SET UP TO MAKE HISTOGRAM BARS
C     QLX(I) = THE NUMBER OF VALUES IN VARIOUS HISTOGRAM BARS.
C     X(I) = THE READ IN VALUES IN  GAUS, BUT COUNT FREQUENCY  IN CHI
C     EX(I)= THE NUMBER OF VALUES EXPECTED IN THESE HISTOGRAM BARS.
C     DX(I) = THE DIFFERENCE BETWEEN ACTUAL AND EXPECTED, 2 DIF. USES
C     DX2(I)= THE SQURE OF DX(I)
C     CHI2(I) = CONTRIBUTIONS TO THE CHI SQUARED CRITERION PARAMETER
C     QLB(I)= THE LOWER BOUNDS OF THE HISTOGRAM BAR INTERVALS
C     QUB(I)= THE UPPER BOUNDS TO THE HISTOGRAM BAR INTERVALS
C     SX = THE SUM OF THE NUMBER OF VALUES IN THE HISTOGRAM BARS
C     SEX= THE SUM OF THE EXPECTED NUMBER OF VALUES IN THE VARIOUS BARS
C     SDX= THE SUM OF THE ABSOLUTE VALUES OF THE VALUES DX(I)
C     SDX2 = THE SUM OF THE VALUES  DX2(I)
C     SCHI2= THE SUM OF THE VALUES CHI2(I),  THE CHI SQUARED SUM
C
C     BY PAUL A. SMITH,  COE COLLEGE, CEDAR RAPIDS, IOWA
C     PLEASE COMMUNICATE ANY DIFFICULTIES TO THE AUTHOR DIRECTLY
C
      DIMENSION X(200),DX(200),QLX(18),EX(18),DX2(18),CHI2(18),QLB(18),
     . QUB(17),LX(17),TITLE(40)
    1 FORMAT(40A2)
    2 FORMAT(1H1,40A2)
    5 READ(2,1) (TITLE(I),I=1,40)
      WRITE(3,2) (TITLE(I),I=1,40)
      K = 200
      RJT = 0.0
      CALL FREE(X,K)
C     TEST TO BE SURE A SET OF VALUES WAS ACTUALLY READ IN
      IF(K) 15,5,15
   15 QK = K
      SX = 0.0
      DO 20 I = 1,K
   20 SX = SX + X(I)
      AVE = SX/QK
      SDX2 = 0.0
      DO 40 I = 1,K
      DX(I) = X(I) - AVE
   40 SDX2 = SDX2 + DX(I)**2
      S = SQRT(SDX2/QK)
C     USE LEAST SQUARES FIT FUNCTION REPRESENTATION OF CHAUVENET CRITERION
C     FUNCTION WHICH IS DEPENDENT ON THE NUMBER OF VALUES INVOLVED
      CHAUV = S*(SQRT(8.39+8.62*ALOG(QK)/2.30259)-2.08-0.111*ALOG(QK)/
     12.30259)
      J = 0
      DO 50 I = 1,K
C     REJECT ALL VALUES EXACTLY EQUAL TO ZERO AS THEY MAY BE KEYPUNCH OR READ
C     ERRORS AND WOULD NOT LIKELY BE REJECTED BY CONSIDERATIONS FOLLOWING
C     VALUES VERY CLOSE TO ZERO ARE PERMITTED, FOR EXAMPLE  0.0001
C     NEGATIVE VALUES ARE PERMITTED
      IF(ABS(X(I))-0.0000001) 42,41,41
C     REJECT VALUES VERY FAR OUT ON TAILS OF DISTRIBUTION
   41 IF(CHAUV-ABS(DX(I))) 42,42,48
   42 IF(RJT) 43,43,45
   43 WRITE(3,44)
   44 FORMAT('0 SORRY, BUT THESE VALUES LOOK OUT OF PLACE AND SO ARE REJE
     .CTED'///11X,'VALUE'8X,'AVERAGE'6X,'DEVIATION  PERMITTED DEV')
      RJT = 1.0
   45 WRITE(3,46) X(I),AVE,DX(I),CHAUV
   46 FORMAT(' '4F15.4)
      GO TO 50
   48 J = J + 1
      DX(J) = DX(I)
   50 CONTINUE
C     END OF EXTREME VALUE REJECTION LOOP
C     TEST TO SEE IF ANY VALUES REJECTED
      IF(K-J) 52,60,52
   52 K = J
C     IF VALUES WERE REJECTED RECOMPUTE THE RAW SET OF VALUES
      DO 55 I = 1,K
   55 X(I) = DX(I) + AVE
C     SINGLE SPACE THE PRINTER
      WRITE(3,46)
C     IF VALUES WERE REJECTED RE-CHECK REMAINING LIST FOR EXTREME VALUES
      GO TO 15
C     THE RAW SET OF VALUES HAS BEEN CLEANED AND WE HAVE AVERAGE AND STANDARD
C     DEVIATION
C     MAKE A STUDY OF  RUNNING AVERAGE AND STANDARD DEVIATIONS AND
C     THEN FOR SUBSETS OF THE FULL SET COMPUTE AVERAGE AND STANDARD DEVIATIONS
   60 CALL SUBSE(X,K)
C     PUT THE VALUES AND THEIR DEPARTURE FROM THE AVERAGE IN NUMERICAL ORDER
      CALL ORDE(X,DX,K)
      WRITE(3,62)  (X(I), DX(I), I=1,J)
   62 FORMAT(///'0 ORDERED VALUES    DEV FROM AVE'//(' '2F15.4))
C     COUNT UP HOW MANY VALUES LANDED IN EACH OF CERTAIN HISTOGRAM BAR INTERVALS
      CALL COUNT(K,DX,SP,INTV,LX,QLB,QUB,AVE)
      DO 70 I = 1,INTV
   70 QLX(I) = LX(I)
C     COMPUTE THE EXPECTED GAUSSIAN DISTRIBUTION OF COUNTS FOR HISTOGRAM BARS
      CALL GEX(INTV,K,EX,SP,S)
C     COMPUTE THE CHI SQUARE VALUE FOR OBSERVED VERSUS EXPECTED DISTRIBUTION
      CALL CHI(INTV,QLX,EX,DX,DX2,CHI2,QLB,QUB,SX,SEX,SDX,SDX2,SCHI2,
     1AVE,S)
C     PERMIT MULTIPLE SETS OF DATA
      GO TO 5
C     WILL CALL EXIT ON ENCOUNTERING A // * MONITOR COMMENT CARD
      END
// DUP
*STORE      WS  UA  GAUS
```

```
      SUBROUTINE SUBSE(X,K)
      DIMENSION X(1)
C     MAKE STUDY OF RUNNING AVERAGE AND STANDARD DEVIATION OF SET OF VALUES
C     DO NOT MAKE STUDY ON A SET OF LESS THAN 9 VALUES
      IF(K-9) 5,10,10
    5 RETURN
   10 WRITE(3,20) X(1),X(1)
   20 FORMAT('0 OUTPUT FROM SUBROUTINE SUBSET'/'0'9X'VALUES',3(8X'RUNNING
     .')/6X,'AS READ IN'8X,'AVERAGE'10X,'SIGMA',3X,'SIGMA OF AVE'/'0'2F1
     .5.4)
      SUM = X(1)
      SUM2 = X(1)**2
C     COMPUTE RUNNING AVERAGE AND STANDARD DEVIATION FOR SET OF VALUES
      DO 100 I = 2,K
      SUM = SUM + X(I)
      SUM2 = SUM2 + X(I)**2
C     AVE  =  AVERAGE OF SET OF VALUES INCLUDED UP TO THIS POINT
C     SDS  =  STANDARD DEVIATION OF THE SET
C     SDA  =  STANDARD DEVIATION OF THE AVERAGE
      AVE = SUM/FLOAT(I)
      SDS = SQRT((SUM2-FLOAT(I)*AVE**2)/FLOAT(I-1))
      SDA = SQRT((SUM2-FLOAT(I)*AVE**2)/FLOAT(I*(I-1)))
      WRITE(3,30) X(I),AVE,SDS,SDA
   30 FORMAT(' '4F15.4)
  100 CONTINUE
      WRITE(3,101) K
  101 FORMAT('0 THERE WERE 'I3,' VALUES IN THE FULL CLEAN SET')
C     PREPARE TO MAKE A STUDY OF SUBSETS OF THE FULL SET OF VALUES
C     NOT ALL VALUES CAN BE USED IN STUDY OF SUBSETS, SEEK TO WASTE AS FEW
      AS IS POSSIBLE
C     COMPUTE APPROPRIATE SIZE OF SUBSETS TO WASTE LEAST NUMBER OF VALUES
C     NGK =   NUMBER OF GROUPS KEEP
C     KQ  =   NUMBER OF VALUES WASTED IN BEST CHOICE OF NG TO DATE
C     N   =   NUMBER OF VALUES TO BE GROUPED IS EQUAL TO NUMBER IN LIST
C     NS  =   SQUARE ROOT OF NUMBER OF VALUES, ROUGH ESTIMATE OF NUMBER GROUPS
C     NI  =   NUMBER OF VALUES EACH SIDE OF  NS  TO BE CONSIDERED
C     NX  =   MAXIMUM VALUE TO BE CONSIDERED IN CONSIDERING NUMBER OF GROUPS
C     NG  =   THE NUMBER OF GROUPS CURRENTLY BEING CONTEMPLATED
      NGK = 0
      KG = 1000
      N = K
      NS = SQRT(FLOAT(N))
      NI = (FLOAT(N))**0.25
      NX = NS + NI
      DO 105 NG = NS,NX
C     CHECK TO SEE HOW MANY VALUES WOULD BE WASTED WITH CURRENT NG
      IF(IABS(N-(N/NG)*NG)-KQ) 104,104,105
C     STORE CURRENT BETTER THAN ANY FORMER VALUE OF NG WITH ASSOCIATED KQ
  104 NGK = NG
      KQ = IABS(N-(N/NG)*NG)
  105 CONTINUE
C     NR   =  NUMBER OF READINGS PER GROUP
C     NU   =  NUMBER OF VALUES USED
C     SUM  =  SUM OF INDIVIDUAL VALUES IN SUBSET
C     SUM2 =  SUM OF SQUARES OF INDIVIDUAL VALUES IN SUBSET
C     GSUM =  GRAND SUM OF SUBSET AVERAGES
```

```
C     GSUM2 = GRAND SUM OF SUBSET AVERAGES SQUARED
C     ZSUM =  SUM OF STANDARD DEVIATIONS OF INDIVIDUAL SUBSETS
C     ZSUM2 = SUM OF SQUARES OF STANDARD DEVIATIONS OF SUBSETS
C     NG   =  NUMBER IN SUBSET GROUP
      NG = NGK
      NR = N/NG
      WRITE(3,106)  NG,NR
  106 FORMAT(//'0 BREAK UP VALUES INTO'I3,' SUBSETS OF 'I3,' VALUES EACH')
      NU = NR*NG
C     SIZE OF GROUP SUBSET COMPUTATION COMPLETED
C     COMPUTE AVERAGE AND STANDARD DEVIATION FOR THE VARIOUS SUBSETS OF VALUES
      SUM = 0
      SUM2 = 0
      GSUM = 0
      GSUM2 = 0
      ZSUM = 0
      ZSUM2 = 0
      NG = 0
      WRITE(3,110)
  110 FORMAT('0'5X,'SUBSET AVE'21X,'ITS SIGMA'/' ')
C     MAKE STUDY OF SUBSETS
      DO 200 I = 1,K
      SUM = SUM + X(I)
      SUM2 = SUM2 + X(I)**2
      IF(I-NR*(I/NR)) 200,120,200
C     NOTE I(MOD NR) HERE
C     COMPLEAT COMPUTATIONS FOR THE GIVEN SUBSET
  120 AVE = SUM/FLOAT(NR)
      SDS = SQRT((SUM2-FLOAT(NR)*AVE**2)/(NR-1))
      GSUM = GSUM + AVE
      ZSUM = ZSUM + SDS
      GSUM2 = GSUM2 + AVE**2
      ZSUM2 = ZSUM2 + SDS**2
      WRITE(3,150) AVE,SDS
  150 FORMAT(' 'F15.4,15X,F15.4)
      NG=NG+1
C     HAS FULL SET OF SUBSETS BEEN CONSIDERED
      IF(I-NU) 180,210,180
C     ZERO ACCUMULATORS FOR SUMS OF VALUES ASSOCIATED WITH SUBSETS
  180 SUM = 0
      SUM2 = 0
  200 CONTINUE
  210 WRITE(3,211)
  211 FORMAT(///5X'AVE OF AVES    THEIR SIGMA  AVE OF SIGMAS    THEIR SI
     .GMA'/' ')
C     COMPUTE GRAND AVERAGES AND STANDARD DEVIATIONS
C     G MEANS GRAND, Z MEANS STANDARD DEVIATION IN FOLLOWING FOUR NAMES
      GAVE = GSUM/FLOAT(NG)
      ZAVE = ZSUM/FLOAT(NG)
      GAVEZ = SQRT((GSUM2-FLOAT(NG)*GAVE**2)/(NG-1))
      ZAVEZ = SQRT((ZSUM2-FLOAT(NG)*ZAVE**2)/(NG-1))
      WRITE(3,30) GAVE,GAVEZ,ZAVE,ZAVEZ
      RETURN
      END
      SUBROUTINE ORDE(X,Y,K)
C     PUTS THE TWO COLUMN VECTORS X(I) AND Y(I) IN ORDER ACCORDING TO THE
C     VALUES IN X(I).  THE PAIR X(J) AND Y(J) ARE KEPT TOGETHER IN ORDERING.
C     THE SMALLEST X(I) IS PUT IN X(1)
      DIMENSION X(1),Y(1)
C                       CHECK EVERY ELEMENT
   30 DO 50 J = 1,K
C                       NOTE LOWER AS INITIAL SMALLEST ELEMENT
      L = J
C                       CHECK EACH ELEMENT HIGHER THAN THE FIRST
      DO 40 I = J,K
C                       IF THE LOWER IS .LE. THE   HIGHER ONE  ALL O.K.
      IF(X(L)-X(I)) 40,40,35
C                       IF HIGHER ELEMENT SMALLER THAN LOWER ONE, NOTE THIS
   35 L = I
C                       L   IS KEPT THE INDEX OF THE SMALLEST ELEMENT EXAMINED
   40 CONTINUE
C                       ON EXIT FROM DO LOOP L  IS INDEX OF SMALLEST ELEMENT
C                       TEMPORARILY STORE VALUES OF LOWEST ELEMENT STUDIED
      TX = X(J)
      TY = Y(J)
C                       TRANSFER SMALLEST ELEMENT TO LOWEST POSITION STUDIED
      X(J) = X(L)
      Y(J) = Y(L)
C                       PLACE TEMPORARY VALUE IN PLACE OF SMALLEST VALUE
      X(L) = TX
   50 Y(L) = TY
      RETURN
      END
      SUBROUTINE COUNT(L,DX,SP,K,LX,QLB,QUB,AVE)
C     THIS SUBROUTINE ACCEPTS  L  VALUES  DX(I).  A HISTOGRAM ANALYSIS
C     IS MADE BY SETTING UP  K  INTERVALS OF WIDTH  SP  WHICH SPAN THE
C     SET OF VALUES INCLUDED IN DX(I), HAVING UPPER AND LOWER BOUNDS
C     STORED IN QUB(I), AND QLB(I).  THE NUMBER OF VALUES DX(I) WHICH ARE
C     FOUND TO FALL IN THE VARIOUS INTERVALS ARE STORED IN LX(I).
C     A HISTOGRAM OF LX(I) IS INCLUDED USING **************
C     THE NUMBER OF HISTOGRAM BARS IS AN ODD NUMBER JUST LESS THAN THE
C     SQUARE ROOT OF THE NUMBER OF VALUES IN THE LIST
      INTEGER STAR,BLANK
      DIMENSION DX(1),LX(1),QLB(1),QUB(1),ID(30)
      BLANK = 16448
      STAR = 23644
      QL = L
      K = SQRT(QL) - 0.5
      IF(K-(K/2)*2) 4,2,4
    2 K = K - 1
    4 IF(K-3) 6,6,8
    6 K = 3
    8 QK = K
      DO 10 I = 1,K
   10 LX(I) = 0
      SP = 1.0001*(DX(L)-DX(1))/QK
      AM = DX(1) + (DX(L) - DX(1))/2.0
      DO 20 I = 1,K
      QI = I - K/2 - 1
      DO 18 J = 1,L
      QLB(I) = (QI-.5)*SP + AM
      QUB(I) = (QI+.5)*SP + AM
      IF(DX(J)-QLB(I)) 18,18,14
   14 IF(DX(J)-QUB(I)) 16,16,18
   16 LX(I) = LX(I) + 1
   18 CONTINUE
   20 CONTINUE
      DO 25 I = 1,30
   25 ID(I) = I
      WRITE(3,31)
   31 FORMAT(///'0'4X,'LOWER BOUND'4X,'UPPER BOUND HISTOGRAM BARS'/' ')
      DO 60 I = 1,K
      LXU = LX(I)
      QLB(I) = QLB(I) + AVE
      QUB(I) = QUB(I) + AVE
      IF(LXU-90) 37,37,36
   35 FORMAT(' ',2F14.4,90A1)
   36 LXU = 90
   37 WRITE(3,35) QLB(I),QUB(I)
      IF(LXU) 55,55,50
   50 WRITE(3,100) (STAR,L = 1,LXU)
   55 QLB(I) = QLB(I) - AVE
   60 QUB(I) = QUB(I) - AVE
  100 FORMAT('+',29X,90A1)
      RETURN
      END
      SUBROUTINE CHI(K,X,EX,DX,DX2,CHI2,QLB,QUB,SX,SEX,SDX,SDX2,SCHI2,
     1AVE,S)
C     CHI IS USED IN CONJUNCTION WITH SUBROUTINE GAUS.  SEE THE LATTER
C     FOR THE SIGNIFICANCE OF THE VARIABLES IN CHI.   SUBROUTINE CHI
C     ACCEPTS K VALUES OF X(I) AND EX(I), NORMALIZES THE VALUES OF
C     EX(I) SO THAT THE SUM OF THEIR VALUES IS EQUAL TO THE SUM OF THE
C     VALUES OF X(I), THEN CALCULATES THE VALUES OF SEX,SDX,SDX2,SCHI2
      DIMENSION X(1),DX(1),EX(1),DX2(1),CHI2(1),QLB(1),QUB(1)
      SX = 0.0
      SEX = 0.0
      DO 10 I = 1,K
      SX = SX + X(I)
   10 SEX = SEX + EX(I)
      DO 15 I = 1,K
   15 EX(I) = (SX/SEX)*EX(I)
      SEX = 0.0
      SDX = 0.0
      SDX2 = 0.0
      SCHI2 = 0.0
      DO 30 I = 1,K
      SEX = SEX + EX(I)
      DX(I) = EX(I) - X(I)
      IF(ABS(DX(I))-1.0E-15) 25,26,26
   25 DX(I) = 1.0E-15
   26 SDX = SDX + ABS(DX(I))
      DX2(I) = DX(I)**2
      SDX2 = SDX2 + DX2(I)
      IF(EX(I)-1.0E-15) 27,28,28
   27 EX(I) = 1.0E-15
   28 CHI2(I) = DX2(I)/EX(I)
   30 SCHI2 = SCHI2 + CHI2(I)
      QK = K
      WRITE(3,100)  (X(I),EX(I),DX(I),DX2(K),CHI2(I),QLB(I),QUB(I),I=1,K)
  100 FORMAT(////' NOW LETS LOOK AT THE EXPECTED HISTOGRAM'//'0  COUNT E
     .XPECT    DIF   DIF2   CHI2  LOWER  UPPER'//(' '7F7.2))
      WRITE(3,120) SX,SEX,SDX,SDX2,SCHI2
  120 FORMAT('0'7F7.2)
      WRITE(3,130)
  130 FORMAT('0 THE LAST LINE OF VALUES GIVES SUMS OF VALUES IN COLUMNS')
      RETURN
      END
      SUBROUTINE GEX(K,L,EX,SP,S)
C     THERE ARE  L  VALUES  HAVING A GAUSSIAN DISTRIBUTION WHICH HAVE
C     BEEN FOUND TO FALL IN  K  INTERVALS  OF WIDTH  SP.  THE VALUE OF
C     K  IS AN ODD INTEGER.  THE STANDARD DEVIATION OF THE  L  VALUES IS
C     THE NUMBER  S.  THIS SUBROUTINE CALCULATES THE VALUES OF EX(I)
C     WHICH ARE PROPORTIONAL TO THE NUMBER OF VALUES EXPECTED TO FALL IN
C     EACH INTERVAL OF WIDTH  SP.  NOTE THAT THE SUM OF THE EX(I) IS NOT
C     EQUAL TO THE VALUE OF  L  BECAUSE THE EXPECTED GAUSSIAN DISTRIBUTION
C     EXTENDS BEYOND THE RANGE COVERED BY THE  K  INTERVALS OF WIDTH  SP
      DIMENSION EX(1)
      QL = L
      DO 50 I = 1,K
      QI = I - K/2 - 1
      IF(K-(K/2)*2) 15,14,15
   14 QI = QI + 0.5
   15 SUM = 0
      DO 40 J = 1,10
      QJ = J
      QJ = QJ*0.1
   40 SUM = SUM + EXP(-((QI-.5+QJ-0.05)*SP)**2/(2.*S**2))
   50 EX(I) = (QL/S/2.5066)*SUM*(SP/10.0)
      RETURN
      END
```

<u>Sample Data Deck for GAUS</u>

```
// XEQ GAUS
DICK ROWE  TORSION PENDULUM PART II
22.2  22.1 22.2 21.9 22.2  21.8 22.0 22.0 22.3 21.9 22.0 22.0 21.8 22.2
22.4 22.4 22.7 22.2 21.5 22.4 21.8 22.1 21.8 22.0 22.0 22.4 21.8 22.2
22.1  22.2 21.7 22.2 22.2 22.2 22.3 22.5 21.6 22.6 22.1 22.0 21.9 21.8
22.0 21.6 22.3 22.5 22.3 22.0 22.3 22.2      9999
```

APPENDIX

Format-Free Input

When one is working with students of little or no previous experience in using a digital computer, one of the major problems encountered is that of their properly entering data on punched cards. This is true both in the case where the computer program is written for the student by the instructor and the case where students write their own programs for simple data analysis. Even for experienced programmers, entering data on cards according to a specified FORMAT can be a serious nuisance. The Coe College FREE STYLE input programs are designed to remove this barrier to easy use of the digital computer; it can easily make the difference between success and total failure in introducing digital computing into instructional activities.

The basic premise of the system is that a student should be able to punch into data cards, in proper sequence, the numbers to be entered into the computer with practically no further restrictions. In general, any recognizable number is legal. In particular:

1. Numbers may be placed anywhere on a card or cards just so long as: a) one or more blanks separate different numbers; b) all of each number is on just one card; and c) numbers from two different "batches" do not appear on the same card.

2. Numbers are read in "batches" of one or more values. Each "batch" corresponds to the execution of a statement of the form CALL FREE(X,NV). This causes up to NV values to be read into the locations X(1), X(2), X(3), . . ., X(NV). Successive cards are searched for numbers until one of the following happens: a) NV values have been read in--RETURN to calling program; b) 9999 encountered on a data card--NV is set equal to the number of values previously read in--RETURN to calling program--"9999" thus means "end of batch" when NV is set big; c) // found in columns one and two--monitor trap.

3. Numbers may contain: a) up to 8 digits; b) algebraic signs may be used freely--the "+" sign is optional; c) a decimal point--optional for integer values; d) a properly positioned "E" in an exponential constant, provided there is no blank space before the "E" and it is followed by an explicit "+" or "-" sign (for example, 0.317E-8).

4. Illegal characters are treated as blanks, _if column one of the card is blank_. This permits the following: a) entries on data cards of the form A = 5.6 so the student may remind himself of the meaning of the number; b) erasure of keypunch errors by overpunching all of the number in question with X's (this greatly speeds up novice use of an electric keypunch and permits ready use of manual inexpensive ones in laboratory situations); c) insertion of pure comment cards and blank cards in the data deck for identification and reminder purposes.

5. Comment cards with a "C" in column one are not examined for numerical values; instead they are printed out on the printer with carriage control according to the content of column two. This permits batched data decks with pagination and student name header printed at the top of his printed output.

The card reading by these programs is slow but very useful with limited amounts of data. Efforts will be made to write faster versions--both with only FORTRAN and also taking advantage of the IDEAL subroutines available on the IBM 1130 which permit a faster search and overlapped I/O.

The following pages contain listings of FREE and associated programs from the FREE STYLE package, as well as a flow chart for FREE, which is the fundamental subroutine. In the flow chart, literals such as "+" within a diamond (decision symbol) indicate a test as to whether the column under consideration contains that symbol. The symbol "b" stands for a blank column.

```
// *   MODIF68  12
// DUP
*DELETE             FREE
// FOR
*ONE WORD INTEGERS
      SUBROUTINE FREE(X,NV)
      DIMENSION X(1),NA(80),PT(10)
C
C     FREE STYLE CARD READER
C
C     CALL FREE(X,NV) WILL READ UP TO NV VALUES INTO X(1) . . . X(NV)
C     THE FIRST ARGUMENT OF THE CALL STATEMENT MUST BE A SINGLY SUBSCRIPTED REAL
C     VARIABLE
C     THE SECOND ARGUMENT OF THE CALL STATEMENT MUST BE AN INTEGER VARIABLE
C     PUNCH VALUES ANYWHERE ON CARD BUT SEPARATE THEM WITH ONE OR MORE BLANKS
C     VALUES MAY HAVE UP TO EIGHT INTEGER DIGITS PUNCHED TO DESIGNATE THEM
C     DECIMAL POINTS ARE HONORED WHERE PUNCHED BUT ARE OPTIONAL
C     VALUES ARE READ OFF CARDS IN ORDER RUNNING FROM LEFT TO RIGHT ON
C     SUCCESSIVE CARDS UNTIL NV VALUES FOUND OR 9999 FOUND
C     YOU MAY PUT AS MANY OR AS FEW VALUES ON EACH CARD AS DESIRED
C     FOUR NINES STANDING ALONE ON CARD SERVE AS SIGN OF END OF VALUES
C     IF  9999  IS ENCOUNTERED NV IS SET TO THE NUMBER OF VALUES PREVIOUSLY READ
C     TO ENTER THE VALUE 9999 INTO THE COMPUTER ENTER IT AS 9999.
C
C     WITH NV = 4 CALL FREE(X,NV) WOULD READ 1 2 3 4 OFF NEXT CARD, LOSE 5 6 7
C      1 2 3 4 5 6 7
C     WITH NV = 10 CALL FREE(X,NV) WOULD READ 1 2 3 4 5 OFF NEXT CARD AND SET
C     NV = 5 BEFORE RETURN FROM THE CALL
C      1 2 3 4 5 9999 6 7 8 9
C
C     LEGAL CHARACTERS ARE 0123456789.E+-
C     THERE MUST BE NO BLANK BEFORE THE  E  AND THERE MUST BE A + OR - AFTER
C     E TYPE NUMBERS ARE HONORED IN FOLLOWING FORMS  12.0E+2 4.5E-2
C     LEGAL VALUES  2 4 5 +23 -24 -23. -.02 +.03 12E+3 -4.5E-23 1E+.01 45632578
C     ILLEGAL VALUES  2.13.52.45.6  23.4E 3  23.8 E+35   9999 (EXCEPT AS END)
C     ILLEGAL VALUES 23E05  E+03  456328754  45.2547865  +36-25+47+89 0.00000005
C
C
C     CARDS HAVING COLUMN ONE BLANK WILL HAVE ALL ILLEGAL CHARACTERS REMOVED
C     BEFORE THE NUMBERS ARE READ OFF IT
C     DATA CARDS WITH ANY PUNCH IN COLUMN ONE EXCEPT C WILL BE READ WITHOUT ANY
C     CHECKING FOR ILLEGAL CHARACTERS
C     THANKS TO ERASURE FEATURE MISPUNCHED CHARACTERS MAY BE 'X-ED' OUT ON CARD
C     THIS MAKES IT  PRACTICAL TO USE MANUAL CARD PUNCHES IN THE LABORATORY
C
C     NOTE THAT COLUMN ONE MUST BE BLANK FOR ERASURE FEATURE TO OPERATE
C     ERASURE OF ILLEGAL CHARACTERS PERMITS DATA CARDS AS THOSE FOLLOWING
C     HERE CONSIDER COLUMN 7 TO BE COLUMN 1 OF A REAL DATA CARD
C      A = 2.3 B = 5.6 C = 4.2 ON FIRST EXPERIMENT 9999
C      FOLLOWING ARE THE TIMES  4 7 9 12 14 16 18 20 9999
C      FOLLOWING ARE THE DISTANCES 5 6 7 9 10 13 45 9999
C     DDATA BY NANCY PHYSICIST ON AUGUST TENTH   (DON'T PUT DIGIT FOR DATE)
C     THIS FEATURE WILL HELP NOVICES KEEP TRACK OF WHAT THE VALUES REPRESENT
C
C     A 'C' IN COLUMN ONE OF A DATA CARD READ BY 'FREE' WILL BE PRINTED ON
```

```
C     THE PRINTER BUT OTHERWISE WILL BE IGNORED
C     THE FEATURE OF A 'C' IN COLUMN ONE PERMITS PRINT-OUT TO BE IDENTIFIED
C     FOR LABORATORY DATA REDUCTION JOBS HAVING MANY SETS OF DATA FOR ONE RUN
C
C     CARDS ARE READ AT BETWEEN 60 AND 120 CARDS PER MINUTE DEPENDING ON CONTENT
C
      NCP = 1
C     NCP = INCREMENT OF COLUMN COUNTER.  SET NCP = 2 FOR ALTERNATE COLUMNS
      NVR = 1
C     NVR = NUMBER OF VALUES READ IN
      PT(1) = 1
C     PT(I) = POWERS OF TEN = 10**(I-1)
      DO 10 NC = 2,10
   10 PT(NC) = PT(NC-1)*10
      NOPX = 1
C     NOPX = NUMBER OF OPERATION IN MULTIPLICATION OF ENTRY VALUES
C     NOPX = 1 MEANS NO OPERATION   NOPX = 2 MEANS MULTIPLY BY POWER OF TEN
   20 READ(2,32) NA
      NANC = NA(1)
      IF(NANC+15552) 40,30,40
C     DATA CARD WITH 'C' IN COLUMN ONE IS PRINTED ON PRINTER
C     'C' CARDS HAVE CARRIAGE CONTROL ON COLUMN 2
C     USE 'C1' TO GO TO TOP OF PAGE, 'C0' FOR DOUBLE SPACE, 'C ' FOR SINGLE
   30 WRITE(3,32) NA(2),NA
   32 FORMAT(81A1)
      GO TO 20
C     TO CUT STORAGE DEMANDS PUT A GO TO 70 STATEMENT AT 40 AND CUT FOLLOWING
   40 IF(NANC-16448) 70,45,70
   45 DO 68 NC = 2,80
      NANC = NA(NC)
      IF(NANC-16448) 50,68,50
C     CHECK TO SEE IF CHARACTER IS A LETTER
   50 IF(NANC+5824) 56,56,51
C     CHECK TO SEE IF CHARACTER IS AN INTEGER
   51 IF(NANC) 68,52,52
C     CHECK TO SEE IF CHARACTER IS A '-'
   52 IF(NANC-24640) 53,68,53
C     CHECK TO SEE IF THE CHARACTER IS A '.'
   53 IF(NANC-19264) 54,68,54
C     CHECK TO SEE IF THE CHARACTER IS AN IBM 026 '+'
   54 IF(NANC-20544) 55,68,55
C     CHECK TO SEE IF THE CHARACTER IS AN IBM 029 '+'
   55 IF(NANC-20032) 56,60,56
C     CHECK TO SEE IF THE CHARACTER IS AN 'E'
   56 IF(NANC+15040) 61,57,61
C     THE 'E' TYPE NUMBERS MUST HAVE A '+' OR '-' FOLLOW THE 'E'
C     CHECK TO SEE IF THE 'E' IS FOLLOWED BY '-' OR '+' OR EITHER TYPE
   57 IF(NA(NC+1)-20544) 58,68,58
   58 IF(NA(NC+1)-20032) 59,68,59
   59 IF(NA(NC+1)-24640) 61,68,61
   60 NA(NC) = 20544
      GO TO 68
   61 NA(NC) = 16448
   68 CONTINUE
   70 NC = 0
C     NC = NUMBER OF COLUMN OF CARD BEING CONSIDERED
   80 NC = NC + NCP
C     LOOK FOR FIRST CHARACTER OF NEXT VALUE
      IF(NC-80) 90,90,20
   90 NANC = NA(NC)
      IF(NANC-16448) 100,80,100
  100 ND = 0
C     ND = NUMBER OF DIGITS IN NUMBER CONSIDERED
      NDD = 1
C     NDD = NUMBER OF DECIMAL DIGITS OF NUMBER CONSIDERED
      NDDF = 0
C     NDDF = INCREMENT OF NDD FOR COUNTING NUMBER OF DIGITS BEYOND '.'
      NVAL = 0
C     NVAL = INTEGER VALUE OF UP TO FIRST FOUR DIGITS OF NUMBER
      NVAL2 = 0
C     NVAL2 = INTEGER VALUE OF DIGITS BEYOND FIRST FOUR DIGITS OF NUMBER
      NC2 = 1
C     NC2 = NUMBER OF COLUMNS BEYOND FOURTH DIGIT OF NUMBER
C     NC2 AND NDD ARE AUGMENTED BY ONE FOR USE IN PT(I)
      NOPXL = NOPX
      NOPX = 1
C     NOPXL = VALUE OF NOPX LAST TIME X(I) HAD VALUE STORED IN IT
      NSG = +1
C     CHECK FOR ALGEBRAIC SIGN OF VALUE
      IF(NANC-24640) 130,120,130
  120 NSG = -NSG
      GO TO 200
C     CHECK FOR BOTH CODES FOR '+'
  130 IF(NANC-20544) 190,200,190
  190 IF(NANC-20032) 220,200,220
  200 NC = NC + NCP
      IF(NC-80) 202,202,340
  202 NANC = NA(NC)
C     CHECK FOR DECIMAL POINT
  220 IF(NANC-19264) 240,230,240
  230 NDDF = 1
      GO TO 200
C     CHECK FOR 'E'
  240 IF(NANC+15040) 260,320,260
C     END OF A VALUE ON DATA CARDS INDICATED BY PRESENCE OF A BLANK
C     CHECK FOR DELIMITER BLANK
  260 IF(NANC-16448) 270,340,270
C     NANC MUST REPRESENT AN INTEGER IF IT REACHES STATEMENT 270
C     FIND THE INTEGER VALUE OF THE CHARACTER IN COLUMN NC
C     THIS PROCEDURE SPECIFIC TO IBM 1130 CHARACTER CODE
C     ANOTHER ROUTINE HERE MAY BE SUBSTITUTED FOR ANOTHER CHARACTER CODE
  270 I = (NANC+4032)/256
C     COUNT NUMBER OF DIGITS AND DECIMAL DIGITS
      ND = ND + 1
      NDD = NDD + NDDF
      IF(ND-5) 275,285,285
  275 NVAL = NVAL*10 + I
      GO TO 200
  285 NVAL2 = NVAL2*10 + I
```

```
      NC2 = NC2 + I
      GO TO 200
  320 NOPX = 2
  340 X(NVR) = ((NVAL*PT(NC2) + NVAL2)/PT(NDD))*NSG
C     NVR = NUMBER OF VALUES READ IN OFF CARD
C     IF THERE WAS AN 'E' DELIMITER ON LAST VALUE MULTIPLY BY POWER OF TEN
C     FRACTIONAL POWERS PERMITTED HERE
C     21.3E+0.5  IS A LEGAL DATA VALUE FOR 'FREE'
      GO TO(360,350),NOPXL
  350 X(NVR-1) = X(NVR-1)*10.0**X(NVR)
      NVR = NVR -1
  360 GO TO(410,480),NOPX
  410 IF(NDDF) 440,430,440
C     FOUR NINES IN A ROW WITHOUT DECIMAL POINT INDICATE END OF STRING OF VALUES
  430 IF(NVAL-9999) 440,450,440
C     READ ONLY UP TO 'NV' VALUES
  440 IF(NVR-NV) 480,460,460
C     DO NOT RETURN THE VALUE 9999
  450 X(NVR) = 0
C     CALL FREE WITH SECOND ARGUMENT A VARIABLE   NEVER A CONSTANT
C     NOTE THAT HERE THE SECOND ARGUMENT HAS ITS VALUE CHANGED
C     NV ON RETURN IS THE NUMBER OF VALUES ACTUALLY READ IN BEFORE 9999
      NV = NVR - 1
  460 RETURN
  480 NVR = NVR + 1
      GO TO 80
      END
// DUP
*STORE      WS  UA  FREE
```

(A Program to Input Parallel Column Vectors)

```
      SUBROUTINE FREXY(X,Y,NV)
C     READS UP TO NV VALUES OF X(I) AND Y(I) IN SEQUENCE  END BY 9999
C     SEE DOCUMENTATION ON FREE
C     IN CALLING PROGRAM X MUST BE DIMENSIONED IN EXCESS OF 2*NV
C     IF DIMENSION X(2*K),Y(K) YOU MAY EQUIVALENCE X(K+1) AND Y(1) TO SAVE CORE
      DIMENSION X(2),Y(1)
C     THE DIMENSION STATEMENT MERELY DECLARES THESE TO BE SUBSCRIPTED VARIABLES
      NV = NV*2
      CALL FREE(X,NV)
      NV = NV/2
      IM = NV - 1
C     IN CASE ONLY ONE PAIR OF VALUES READ IN
      Y(1) = X(2)
      IF(IM) 500,500,20
C     SHIFT EVEN NUMBERED LOCATIONS UP AND ODD NUMBERED LOCATIONS TO BOTTOM
   20 DO 100 I = 1,IM
      I2 = 2*I
      I2P = I2 + 1
C     FOR EACH EVEN NUMBERED LOCATION SHIFT NEXT ODD ONE OUT TO TEMP FIRST
C     THEN SHIFT ALL ORIGINAL EVEN LOCATIONS BELOW IT UP ONE LOCATION
      TEMP = X(I2P)
      DO 80 J = 1,I
      K = I2P - J
      X(K+1) = X(K)
   80 CONTINUE
C     PUT THE ODD LOCATION IN PLACE MADE AT BOTTOM OF EVEN LOCATION STRING
      X(I+1) = TEMP
  100 CONTINUE
C     PUT ALL EVEN LOCATIONS INTO THE Y COLUMN VECTOR
      DO 200 N = 1,NV
      K = NV + N
  200 Y(N) = X(K)
  500 RETURN
      END
```

Subroutines That Complement Free By Facilitating Reading In Unscripted Values

```
SUBROUTINE RD1(A)
DIMENSION X(1)
NV = 1
CALL FREE(X,NV)
A = X(1)
RETURN
END
```

```
SUBROUTINE RD2(A,B)
DIMENSION X(2)
NV = 2
CALL FREE(X,NV)
A = X(1)
B = X(2)
RETURN
END
```

```
SUBROUTINE RD3(A,B,C)
DIMENSION X(3)
NV = 3
CALL FREE(X,NV)
A = X(1)
B = X(2)
C = X(3)
RETURN
END
```

```
SUBROUTINE IRD1(NA)
DIMENSION X(1)
NV = 1
CALL FREE(X,NV)
NA = X(1) +0.0001
RETURN
END
```

```
SUBROUTINE IRD2(NA,NB)
DIMENSION X(2)
NV = 2
CALL FREE(X,NV)
NA = X(1) +0.0001
NB = X(2) +0.0001
RETURN
END
```

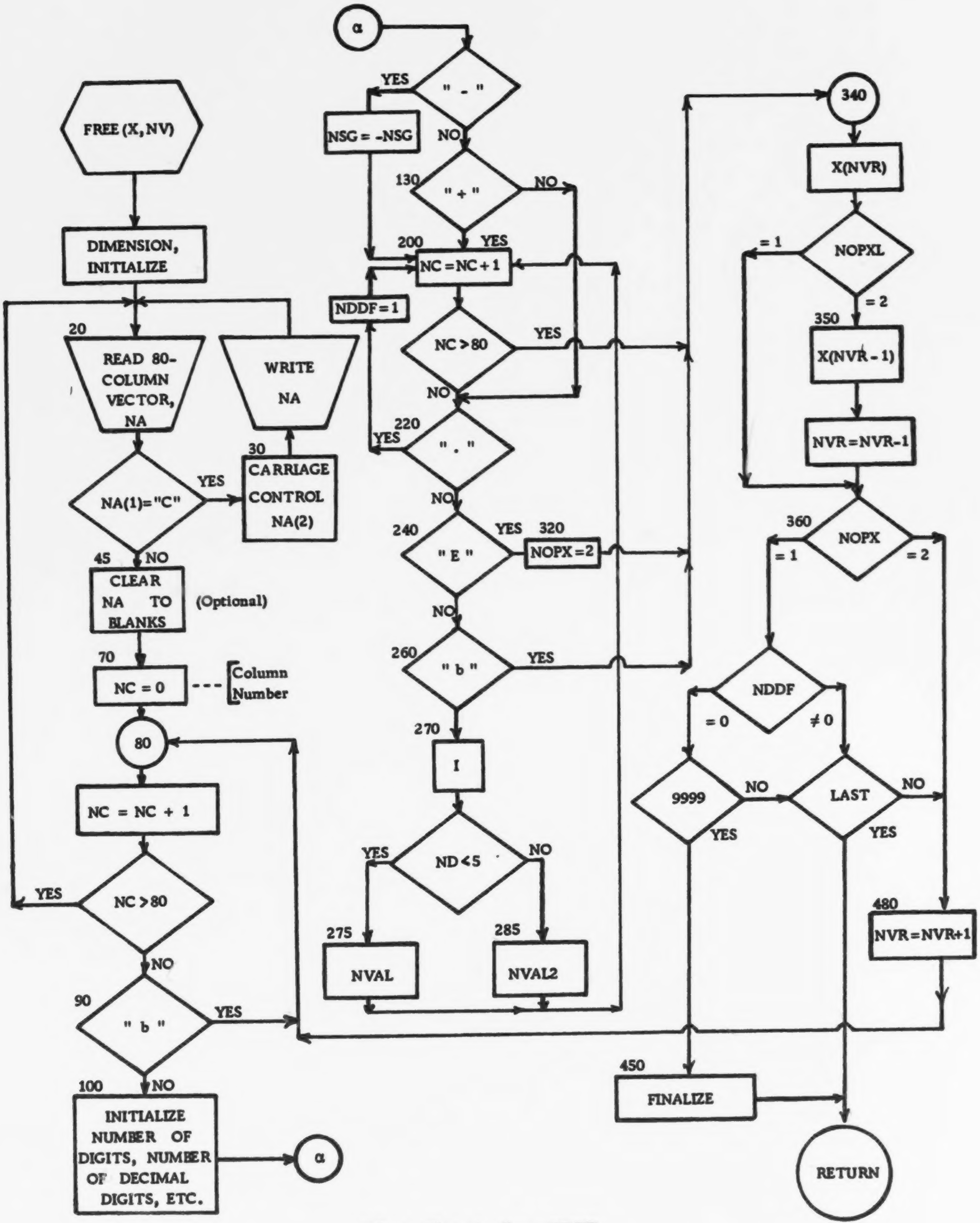
```
SUBROUTINE IRD3(NA,NB,NC)
DIMENSION X(3)
NV = 3
CALL FREE(X,NV)
NA = X(1) +0.0001
NB = X(2) +0.0001
NC = X(3) +0.0001
RETURN
END
```

(Sample Print-Out From "GAUS")

```
DICK ROWE  TORSION PENDULUM PART II

OUTPUT FROM SUBROUTINE SUBSET

        VALUES          RUNNING          RUNNING          RUNNING
     AS READ IN         AVERAGE            SIGMA     SIGMA OF AVE

        22.2000         22.2000
        22.1000         22.1500           0.0704           0.0498
        22.2000         22.1666           0.0596           0.0344
        21.9000         22.0999           0.1419           0.0709
        22.2000         22.1199           0.1305           0.0583
        21.8000         22.0666           0.1757           0.0717
        22.0000         22.0571           0.1622           0.0613←
        22.0000         22.0499           0.1513           0.0535
        22.3000         22.0777           0.1646           0.0548
        21.9000         22.0599           0.1648           0.0521
        22.0000         22.0545           0.1575           0.0475
        22.0000         22.0499           0.1514           0.0437
        21.8000         22.0307           0.1604           0.0445
        22.2000         22.0428           0.1611           0.0430
        22.4000         22.0666           0.1809           0.0467
        22.4000         22.0874           0.1938           0.0484
        22.7000         22.1235           0.2395           0.0580
        22.2000         22.1277           0.2334           0.0550
        21.5000         22.0947           0.2686           0.0616
        22.4000         22.1099           0.2703           0.0604
        21.8000         22.0952           0.2720           0.0593
        22.1000         22.0954           0.2653           0.0565
        21.8000         22.0825           0.2665           0.0555
        22.0000         22.0791           0.2611           0.0533
        22.0000         22.0759           0.2561           0.0512
        22.4000         22.0884           0.2592           0.0508
        21.8000         22.0777           0.2605           0.0501
        22.2000         22.0821           0.2569           0.0485
        22.1000         22.0827           0.2520           0.0468
        22.2000         22.0866           0.2488           0.0454
        21.7000         22.0741           0.2543           0.0456
        22.2000         22.0781           0.2511           0.0444
        22.2000         22.0817           0.2485           0.0432
        22.2000         22.0852           0.2454           0.0420
        22.3000         22.0914           0.2449           0.0413
        22.5000         22.1027           0.2505           0.0417
        21.6000         22.0891           0.2608           0.0428
        22.6000         22.1026           0.2701           0.0438
        22.1000         22.1025           0.2662           0.0426
        22.0000         22.0999           0.2633           0.0416
        21.9000         22.0950           0.2620           0.0409
        21.8000         22.0880           0.2630           0.0405
        22.0000         22.0860           0.2603           0.0396
        21.6000         22.0749           0.2672           0.0402
```

```
         22.3000          22.0799          0.2665          0.0397
         22.5000          22.0891          0.2706          0.0399
         22.3000          22.0935          0.2698          0.0393
         22.0000          22.0916          0.2672          0.0385
         22.3000          22.0958          0.2663          0.0380
         22.2000          22.0979          0.2643          0.0373

 THERE WERE  50 VALUES IN THE FULL CLEAN SET


BREAK UP VALUES INTO  7 SUBSETS OF  7 VALUES EACH

     SUBSET AVE                         ITS SIGMA

         22.0571                           0.1622
         22.0285                           0.1709
         22.1999                           0.4125
         22.0428                           0.2152
         22.1285                           0.1979
         22.0714                           0.3640
         22.1428                           0.2992


    AVE OF AVES     THEIR SIGMA  AVE OF SIGMAS     THEIR SIGMA

         22.0959          0.0635          0.2603          0.0991



 ORDERED VALUES   DEV FROM AVE

         21.5000         -0.5979
         21.6000         -0.4979
         21.6000         -0.4979
         21.7000         -0.3979
         21.8000         -0.2979
         21.8000         -0.2979
         21.8000         -0.2979
         21.8000         -0.2979
         21.8000         -0.2979
         21.8000         -0.2979
         21.9000         -0.1979
         21.9000         -0.1979
         21.9000         -0.1979
         22.0000         -0.0979
         22.0000         -0.0979
         22.0000         -0.0979
         22.0000         -0.0979
         22.0000         -0.0979
         22.0000         -0.0979
         22.0000         -0.0979
         22.0000         -0.0979
         22.0000         -0.0979
         22.1000          0.0020
```

```
      22.1000           0.0020
      22.1000           0.0020
      22.1000           0.0020
      22.2000           0.1020
      22.2000           0.1020
      22.2000           0.1020
      22.2000           0.1020
      22.2000           0.1020
      22.2000           0.1020
      22.2000           0.1020
      22.2000           0.1020
      22.2000           0.1020
      22.2000           0.1020
      22.2000           0.1020
      22.3000           0.2020
      22.3000           0.2020
      22.3000           0.2020
      22.3000           0.2020
      22.3000           0.2020
      22.4000           0.3020
      22.4000           0.3020
      22.4000           0.3020
      22.4000           0.3020
      22.5000           0.4020
      22.5000           0.4020
      22.6000           0.5020
      22.7000           0.6020


   LOWER BOUND    UPPER BOUND HISTOGRAM BARS

      21.4999        21.7399 ****
      21.7399        21.9799 *********
      21.9799        22.2200 ************************
      22.2200        22.4600 *********
      22.4600        22.7000 ****


NOW LETS LOOK AT THE EXPECTED HISTOGRAM


  COUNT EXPECT    DIF   DIF2   CHI2  LOWER  UPPER

   4.00   3.69  -0.30   0.09   0.02  -0.59  -0.35
   9.00  12.21   3.21  10.33   0.84  -0.35  -0.11
  24.00  18.18  -5.81  33.75   1.85  -0.11   0.12
   9.00  12.21   3.21  10.33   0.84   0.12   0.36
   4.00   3.69  -0.30   0.09   0.02   0.36   0.60

  50.00  50.00  12.85  54.61   3.59

THE LAST LINE OF VALUES GIVES SUMS OF VALUES IN COLUMNS
```


FREE (X, NV)
DIMENSION, INITIALIZE
20
READ 80-COLUMN VECTOR, NA
WRITE NA
30
CARRIAGE CONTROL NA(2)
NA(1)="C"
YES
45
NO
CLEAR NA TO BLANKS
(Optional)
70
NC = 0
Column Number
80
NC = NC + 1
NC > 80
YES
NO
90
" b "
YES
100
NO
INITIALIZE NUMBER OF DIGITS, NUMBER OF DECIMAL DIGITS, ETC.
α
α
" - "
YES
NSG = -NSG
NO
130
" + "
NO
200
YES
NC = NC + 1
NDDF = 1
NC > 80
YES
NO
220
YES
" . "
NO
240
" E "
YES
320
NOPX = 2
NO
260
" b "
YES
270
I
ND < 5
YES
NO
275
NVAL
285
NVAL2
340
X(NVR)
NOPXL
= 1
= 2
350
X(NVR - 1)
NVR = NVR-1
360
NOPX
= 1
= 2
NDDF
= 0
≠ 0
9999
NO
YES
LAST
NO
YES
480
NVR = NVR+1
450
FINALIZE
RETURN

Flow Chart for FREE

HARMONIC MOTION

Anton F. Vierling

United States Naval Academy
Annapolis, Maryland

INTRODUCTION

The integration of physics concepts with computer programming skills has enabled students with rudimentary mathematical background to solve problems of increasing complexity. Both science and non-science students, when required to solve physics problems by the application of computer techniques, appear to grasp, not only the basic concepts of physics, but the general capabilities and limitations of the computer as well.

The two sample problems which follow can be solved by students who have had only an elementary introduction to analytic geometry and differential calculus. The discussion of these problems is in two parts: part one deals with the description of simple harmonic motion; part two deals with damped simple harmonic motion. Experimentally, students in the second year of the science curriculum at the Naval Academy were found to spend two to three terminal hours on this material after having had one hour of experience with the BASIC programming language and an introductory lecture on the physics involved.

Simple Harmonic Motion

Any motion that repeats itself in equal intervals of time is called periodic, or harmonic. If a body moves about an equilibrium position due to a force that is proportional to the distance from the equilibrium position to the body itself, then the body is said to undergo simple harmonic motion. An ideal example of such a system is a block set on a frictionless plane and attached to a spring. (See Figure 1.) The force on the block due to the spring is always such as to pull the block back to its equilibrium position, ($X = 0$) and is properly called the restoring force. At equilibrium, of course, the force of the spring on the block is equal to zero. At any particular instant, the restoring force is

$$F = -K*X$$

where the asterisk (*) indicates multiplication. The minus sign in the equation shows that this force will always be directed opposite to the displacement in the X-direction.

If Newton's Second Law, $F = M*A$, is applied to the motion of the block of mass M, and if the force F is replaced by the expression $-K*X$, the following relationship results:

$$-K*X = M*A = M*(d^2X/dT^2)$$

since A (acceleration) is the second derivative of X with respect to time, T. Thus

$$M*(d^2X/dT^2) + K*X = 0 \qquad (1.1)$$

which describes the periodic motion of the block-spring system.

From this differential equation of motion, the problem is to determine the position of the mass at every instant of time after the system has been given an initial displacement. Equation 1.1 is a second-order differential equation for which a numerical solution can be obtained if a set of initial conditions are known. For instance, in this example, it will be assumed that at time $T = 0$, the instantaneous velocity of the block (dX/dT) is zero, and the displacement $x = 25$ meters. (For idealized experiments, the cost of equipment is no object.)

The first step in the solution of this problem is to re-write Equation 1.1 as a system of first-order differential equations. The solution of the first-order equations is relatively simple as long as one remembers that the expression dX/dT is the slope of the X versus T curve at any instant of time. To re-write the

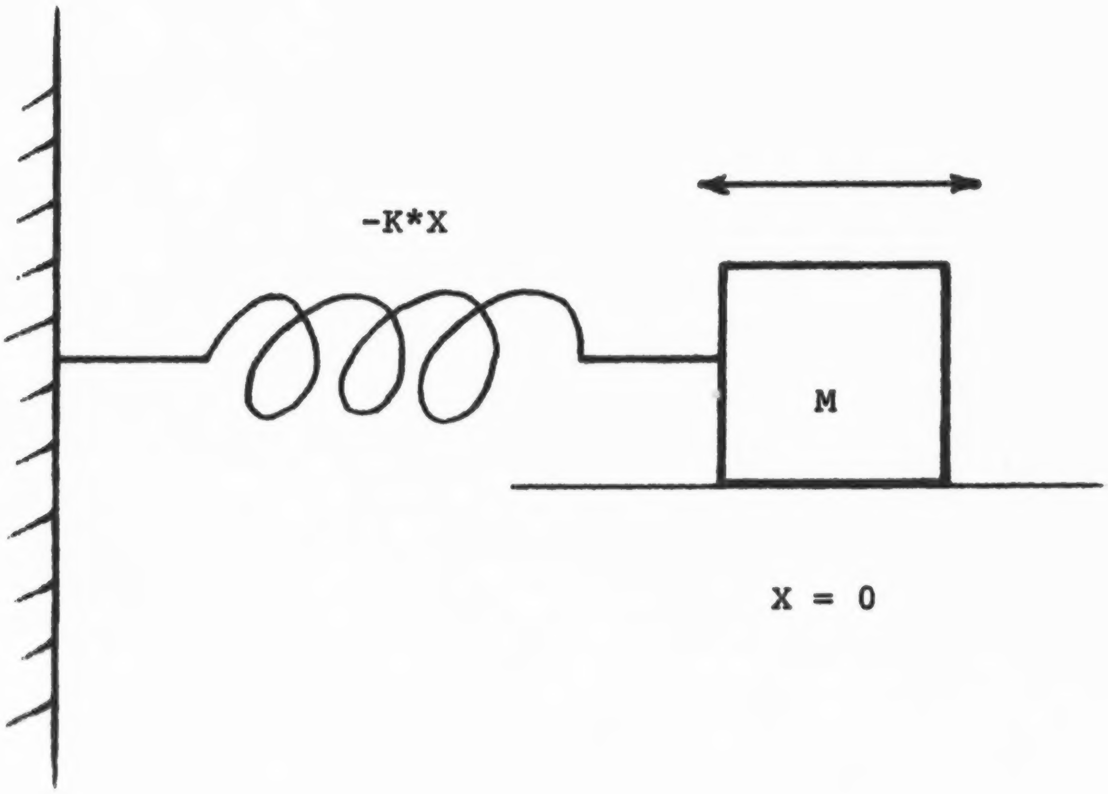


Figure 1

above second-order equations, recall the following definitions of instantaneous velocity and acceleration:

$$V = dX/dT \quad ; \quad A = dV/dT = d^2X/dT^2$$

It is then possible to re-write Equation 1.1 as follows:

$$M*(dV/dT) + K*X = 0$$

or

$$dV/dT = -K*X/M \tag{1.2}$$

and

$$dX/dT = V \tag{1.3}$$

The problem can now be solved approximately if the incremental changes in velocity (ΔV) and displacement (ΔX) which occur during small changes in time (ΔT) can be approximated. In other words, after a small change in time, (ΔT in BASIC will be called D), the new displacement and velocity can be found by

$$X = X + \Delta X$$

$$V = V + \Delta V$$

$$A = -K*X/M$$

These equations are in the form of "assignment statements," $X = X + \Delta X$, which direct the computer to calculate a new value of X by adding ΔX to the current value.

A crude first-order method for approximating the change in displacement and velocity would be

$$\Delta V = A*D$$

$$\text{(change in velocity)} = \text{(acceleration)}*\text{(change in time)}$$

$$\Delta X = V*D$$

$$\text{(change in displacement)} = \text{(velocity)}*\text{(change in time)}$$

where A and V represent the acceleration and velocity respectively at the beginning of the time interval. Therefore,

$$X = X + V*D$$

$$V = V + A*D$$

$$A = -K*X/M$$

A much better estimate could be made by using a weighted average of A and V which could be obtained from values at the beginning, midpoint, and end of the time interval. (See Figures 2a and 2b.) In this figure, the initial slopes

$$X1 = X$$

$$V1 = V$$

$$A1 = -K*X1/M$$

are used to predict the slopes, V2 and A2, at the midpoint of the time interval, $\Delta T = D$. Curved arrows indicate slopes of tangents to the X-T and V-T graphs,

$$\begin{aligned} X2 &= X1 + V1*D/2 \\ V2 &= V1 + A1*D/2 \\ A2 &= -K*X2/M \end{aligned}$$

Next, the values of V2, A2 are used to make a second prediction of the midpoint values of the slopes V3 and A3 starting from X1 and V1. (See Figures 3a and 3b.)

$$\begin{aligned} X3 &= X1 + V2*D/2 \\ V3 &= V1 + A2*D/2 \\ A3 &= -K*X3/M \end{aligned}$$

Finally, this set of slopes is used to predict the values of the slopes V4 and A4 at the end of the time interval (D), starting from X1 and V1. (See Figures 4a and 4b.)

$$\begin{aligned} X4 &= X1 + V3*D \\ V4 &= V1 + A3*D \\ A4 &= -K*X4/M \end{aligned}$$

The theory of the Runge-Kutta approximation shows that if the slopes are weighted by factors 1/6, 2/6, 2/6 and 1/6, respectively, then the approximation will be fourth-order; i.e., correct to within errors proportional to D^5. Thus, if we set

$$\begin{aligned} V &= (V1 + 2*V2 + 2*V3 + V4)/6 \\ A &= (A1 + 2*A2 + 2*A3 + A4)/6 \end{aligned}$$

then the new values of displacement and velocity are found by using the weighted averages as follows:

$$\begin{aligned} X &= X + V*D = X + (V1 + 2*V2 + 2*V3 + V4)*D/6 \\ V &= V + A*D = V + (A1 + 2*A2 + 2*A3 + A4)*D/6 \end{aligned}$$

Note that the values of X1, X2, X3 and X4 were computed in order to obtain the corresponding slopes A1, A2, A3 and A4 of the V-T curve, which in turn predicts the next value of V which predicts the next value of X and so on.

The above algorithms have been incorporated into the computer program SIMPLE HARMONIC MOTION for the determination of the displacement, velocity and acceleration at equal increments of time after the mass is set in motion. This program is designed to give graphic output of X(T) directly, however, the PRINT statement, number 300, can be changed to

```
300 PRINT T, X, V
```

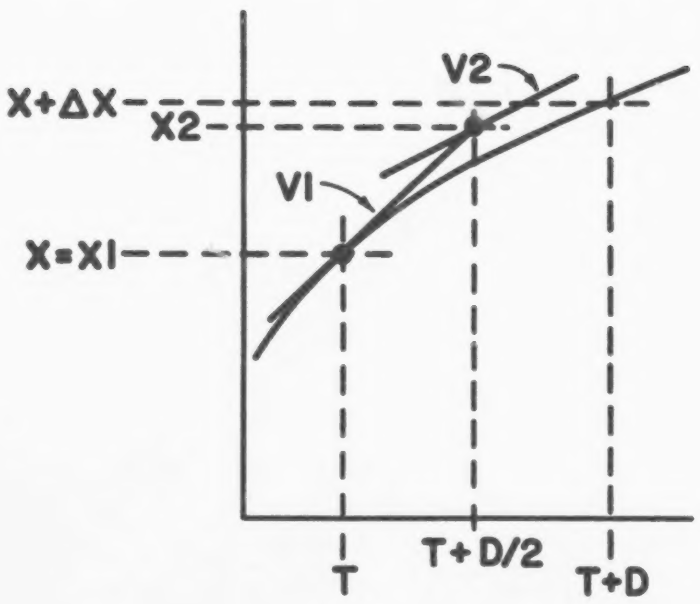

X+ΔX
X2
V2
V1
X=X1
T
T+D/2
T+D

Figure 2-a

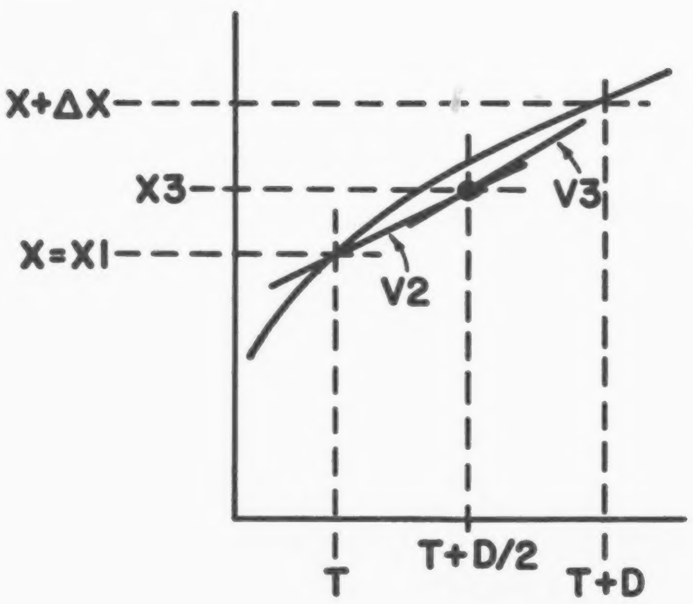

X+ΔX
X3
V3
X=X1
V2
T
T+D/2
T+D

Figure 3-a

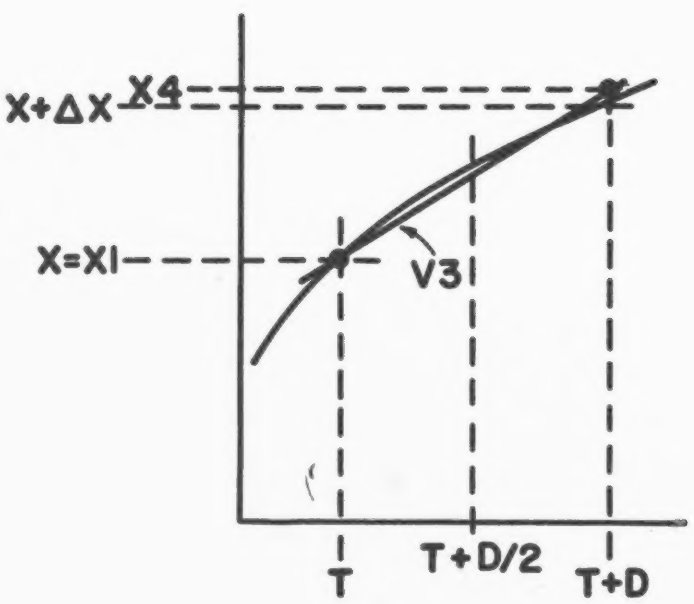

X4
X+ΔX
X=X1
V3
T
T+D/2
T+D

Figure 4-a

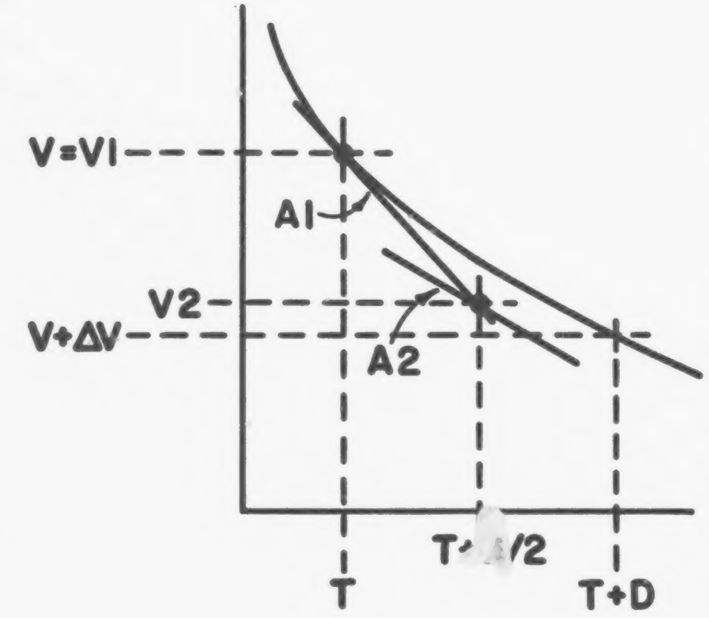

V=V1
A1
V2
V+ΔV
A2
T
T+D/2
T+D

Figure 2-b

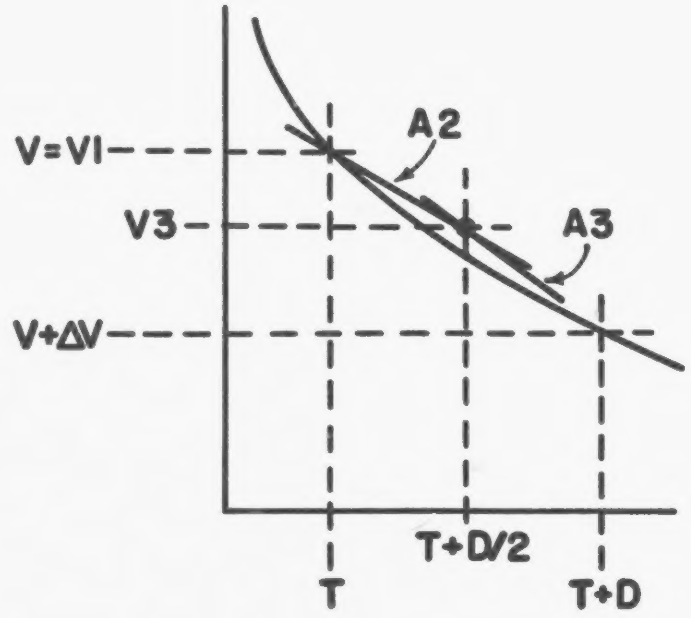

V=V1
A2
V3
A3
V+ΔV
T
T+D/2
T+D

Figure 3-b

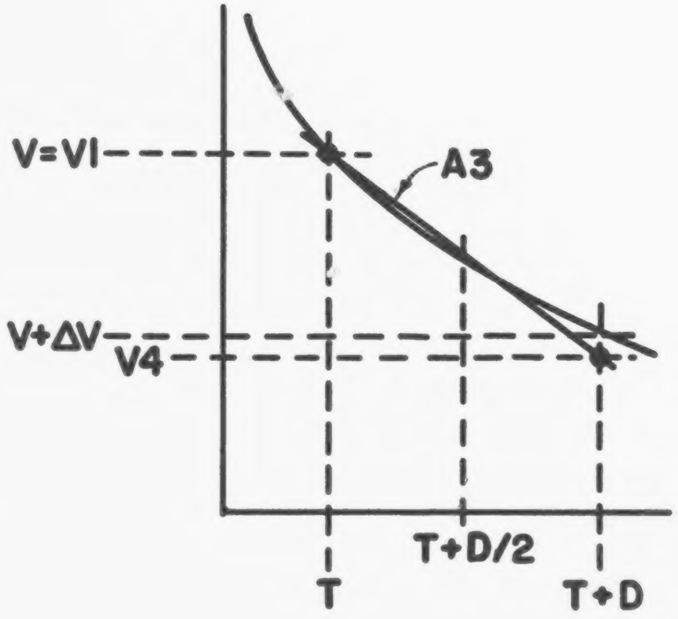

V=V1
A3
V+ΔV
V4
T
T+D/2
T+D

Figure 4-b

for numerical output. T0, X0, and V0 are initial values of T, X and V.

```
100 REM *************** SIMPLE HARMONIC MOTION ***************
120 REM
130 REM *  *  *  *  *  *  *  *  *  *  *  *  *  *  * INPUT DATA
140 REM
150 READ K, D, T0, X0, V0, M
160 DATA 2.8, 0.1, 0, 25, 0, 1
170 PRINT "SPRING CONSTANT = "K"NEWTONS/METER"
180 REM
190 REM *  *  *  *  *  *  *  *  *  *  *  *  INITIAL CONDITIONS
200 REM
210 LET T = T0
220 LET X = X0
230 LET V = V0
240 LET N = 0
250 REM
260 REM *  *  *  *  *  *  *  *  *  *  *  *  *  *  PLOT ROUTINE
270 REM
280 PRINT "                        X - DISPLACEMENT IN METERS"
290 PRINT
300 PRINT TAB(33); "0"
310 PRINT "SECONDS ------------------------------------------";
320 PRINT "------------------"
330 LET Y1 = INT(X + .5) + 36
340 PRINT T; TAB(Y1); "*"
350 LET N = N + 1
360 IF N = 40 THEN 580
370 REM
380 REM *  *  *  *  *  *  *  *  *  *  *  *  * CALCULATE SLOPES
390 LET X1 = X
400 LET V1 = V
410 LET A1 = -K*X1/M
420 LET X2 = X1 + V1*D/2
430 LET V2 = V1 + A1*D/2
440 LET A2 = -K*X2/M
450 LET X3 = X1 + V2*D/2
460 LET V3 = V1 + A2*D/2
470 LET A3 = -K*X3/M
480 LET X4 = X1 + V3*D
490 LET V4 = X1 + A3*D
500 LET A4 = -K*X4/M
510 REM *  *  *  *  *  *  *  *  *  * NEW VALUES OF X, V, AND A
520 LET X = X + (V1 + 2*V2 + 2*V3 + V4)*D/6
530 LET V = V + (A1 + 2*A2 + 2*A3 + A4)*D/6
540 LET A = -K*X/M
550 LET T = T + D
560 SETDIGITS(3)
570 GO TO 330
580 END
```

SPRING CONSTANT = 2.8 NEWTONS/METER

TIME (SECS.)	X-DISPLACEMENT (METERS)	X-VELOCITY (METERS/SECOND)
0	30	0
.05	29.9	-4.2
.1	29.6	-8.36
.15	29.1	-12.5
.2	28.3	-16.5
.25	27.4	-20.4
.3	26.3	-24.2
.35	25.	-27.7
.4	23.5	-31.1
.45	21.9	-34.3
.5	20.1	-37.3
.55	18.2	-39.9
.6	16.1	-42.3
.65	13.9	-44.5
.7	11.7	-46.2
.75	9.32	-47.7
.8	6.9	-48.9
.85	4.44	-49.6
.9	1.94	-50.1
.95	-.566	-50.2
1.	-3.07	-49.9
1.05	-5.55	-49.3
1.1	-8.	-48.4
1.15	-10.4	-47.1
1.2	-12.7	-45.5
1.25	-14.9	-43.5
1.3	-17.1	-41.3
1.35	-19.1	-38.8
1.4	-20.9	-36.
1.45	-22.6	-32.9

```
SPRING CONSTANT  =  2.8 NEWTONS/METER
                        X - DISPLACEMENT IN METERS
                                 0
SECONDS ---------------------------------------------------------------
 0                                                            *
 .1                                                           *
 .2                                                          *
 .3                                                        *
 .4                                                      *
 .5                                                   *
 .6                                               *
 .7                                            *
 .8                                        *
 .9                                    *
 1.                               *
 1.1                          *
 1.2                      *
 1.3                   *
 1.4                *
 1.5             *
 1.6           *
 1.7         *
 1.8        *
 1.9        *
 2.          *
 2.1          *
 2.2            *
 2.3              *
 2.4                 *
 2.5                    *
 2.6                        *
 2.7                            *
 2.8                                *
 2.9                                    *
 3.                                          *
 3.1                                            *
 3.2                                                *
 3.3                                                   *
 3.4                                                      *
 3.5                                                        *
 3.6                                                         *
 3.7                                                          *
 3.8                                                          *
 3.9                                                         *
```

```
SPRING CONSTANT  =  5.6 NEWTONS/METER
                        X - DISPLACEMENT IN METERS
                                 0
SECONDS ---------------------------------------------------------------
 0                                                           *
 .1                                                         *
 .2                                                       *
 .3                                                    *
 .4                                                *
 .5                                          *
 .6                                     *
 .7                               *
 .8                         *
 .9                    *
 1.               *
 1.1           *
 1.2        *
 1.3       *
 1.4       *
 1.5         *
 1.6            *
 1.7                *
 1.8                     *
 1.9                           *
 2.                                  *
 2.1                                      *
 2.2                                            *
 2.3                                                 *
 2.4                                                     *
 2.5                                                       *
 2.6                                                         *
 2.7                                                         *
 2.8                                                        *
 2.9                                                     *
 3.                                                  *
 3.1                                            *
 3.2                                       *
 3.3                                 *
 3.4                           *
 3.5                      *
 3.6                 *
 3.7            *
 3.8         *
 3.9       *
```

Damped Harmonic Motion

In a real spring problem, such as one may study in the laboratory, the amplitude of oscillation gradually decreases to zero. This is, of course, due to friction, and can be accounted for by the addition of damping forces. The oscillations that result are called damped harmonic motion. In Figure 5 friction has been simulated by taking a disk, which is attached to the spring-mass system, and immersing it in a fluid. As a first approximation, one could assume that the faster the disk moves, the greater will be the force of friction, or, in equation form:

friction force = -(constant)*(velocity in X-direction)

or, $F = -B*(dX/dT)$

The minus sign indicates that the friction force will be in a direction opposite to the direction of motion at any instant of time.

In the free-body diagram, it is clear that both the spring force (-K*X) and the friction force (-B*(dX/dT)) act upon the mass, M, at all times to produce the resulting damped harmonic motion. Again, beginning with Newton's Second Law,

$$F = M*A$$
$$-K*X - B*(dX/dT) = M*(d^2X/dT^2)$$

dividing through this equation by M, and rearranging terms, yields

$$d^2X/dT^2 + B/M*(dX/dT) + K/M*X = 0 \qquad (2.1)$$

(Note: the computer reads K/M*X as (K/M)*X.) Equation 2.1 is a second-order differential equation which can be very effectively solved by numerical methods if proper initial conditions are given. As in the previous problem, Equation 2.1 can be transformed to a combination of first-order differential equations. First, by definition, $dX/dT = V$, velocity, and $dV/dT = d^2X/dT^2 = A$, so that

$$dV/dT = -B/M*V - K/M*X \qquad (2.2)$$

In this manner, the problem has been reduced to one which contains two first-order differential equations which can be solved, as before, by a series of approximations based on initial conditions and estimates of the slopes at each increment of time ΔT (in BASIC ΔT will be replaced by D).

Therefore, after each increment of time,

$$X = X + (V1 + 2*V2 + 2*V3 + V4)*D/6 \qquad (2.3)$$

and $$V = V + (A1 + 2*A2 + 2*A3 + A4)*D/6$$

and, from Equation 2.2,

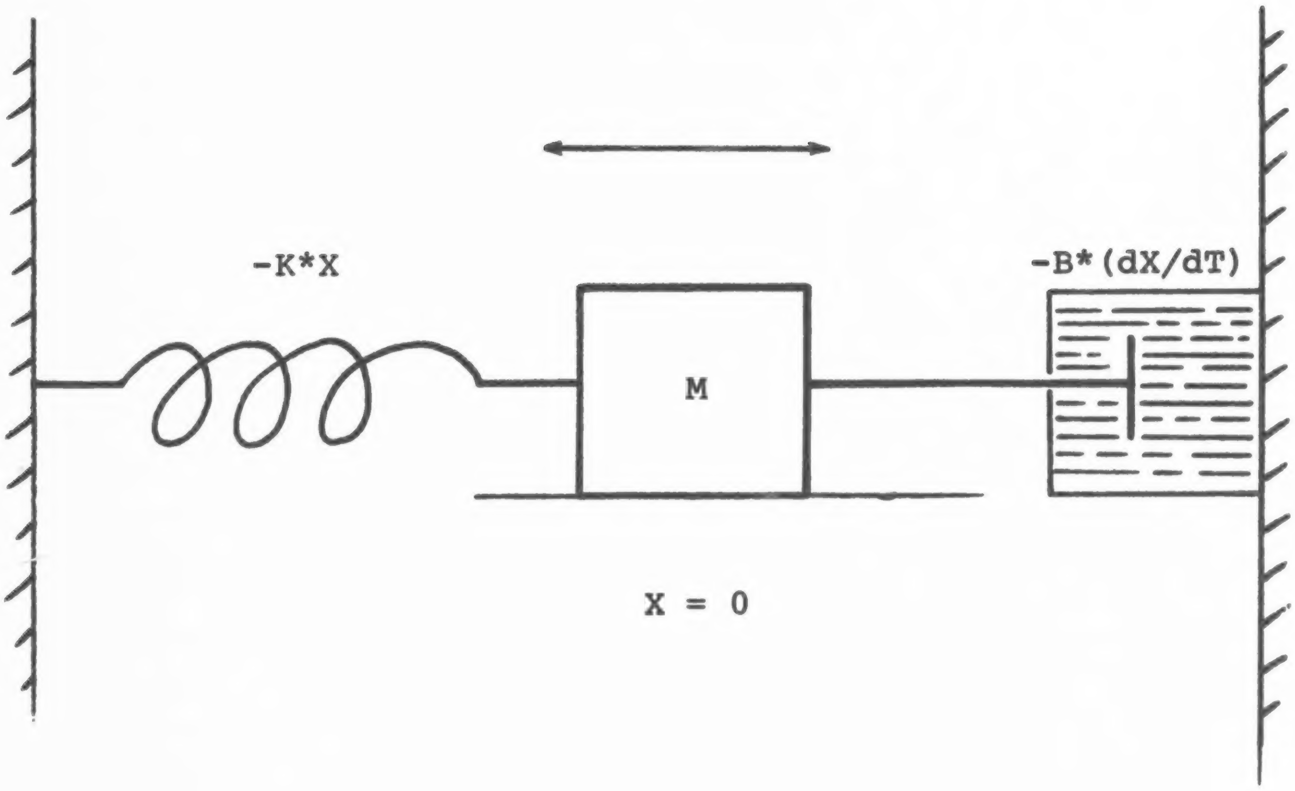

-K*X
M
-B*(dX/dT)
X = 0

Figure 5

```
A  =  -B/M*V - K/M*X
```

The slopes, V1 through V4 and A1 through A4 (velocities and accelerations, respectively) at the four points shown in Figures 2, 3, and 4, are found as follows:

```
(let  C1  =  -B/M  and  C2  =  -K/M)
      X1  =  X
      V1  =  V
      A1  =  C1*V1 + C2*X1 (from Equation 2.2)
```

These slopes predict the slopes at the midpoint of the time increment, D, by

```
X2  =  X1 + V1*D/2
V2  =  V1 + A1*D/2
A2  =  C1*V2 + C2*X2
```

A second estimate of the slopes at the midpoint is made by

```
X3  =  X1 + V2*D/2
V3  =  V1 + A2*D/2
A3  =  C1*V3 + C2*X3
```

and, finally, the slopes at the end of the time interval are found by

```
X4  =  X1 + V3*D
V4  =  V1 + A3*D

A4  =  C1*V4 + C2*X4
```

The only major difference between this and the undamped case is in the calculation of A. Substitution into Equation 2.3 then yields the desired values of X and V.

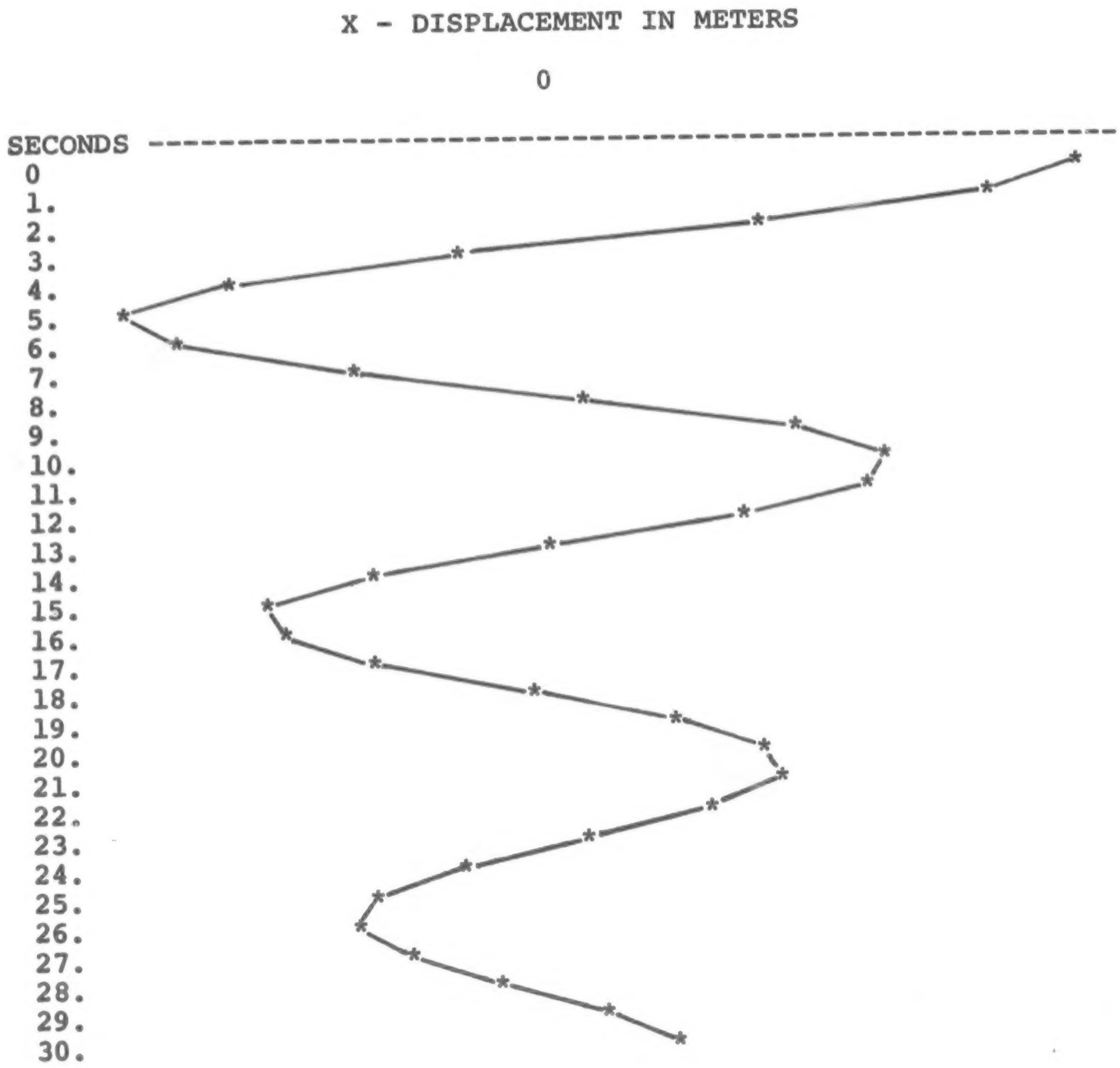
The DAMPED HARMONIC MOTION program which follows illustrates the use of these algorithms to describe the motion of the system. For the sake of clarity, the asterisks in the graphic output have been connected by hand-drawn straight lines. You can easily compare the results for different choices of parameters by superimposing one graph over another on a light-box or against a windowpane.

```
100 REM *************** DAMPED HARMONIC MOTION ***************
110 REM
120 REM * * * * * * * * * * * * * * * INPUT DATA
130 REM
140 READ K, B, M, D, V0, T0, X0
150 DATA 2.8, 0.4, 5, 0.10, 0, 0, 30
160 PRINT
170 PRINT "SPRING CONSTANT = "K"NEWTONS/METER"
180 PRINT
190 PRINT "DAMPING COEFFICIENT = "B"NEWTONS/METER/SEC"
200 REM
210 REM * * * * * * * * * * * *  INITIAL CONDITIONS
220 REM
230 LET C1 = -B/M
240 LET C2 = -K/M
250 LET X = X0
260 LET V = V0
270 LET T = T0
280 REM * * * * * * * * * * *  PRINT COMPUTER OUTPUT
290 PRINT
300 PRINT "TIME","X-DISPLACEMENT", , "VELOCITY"
310 LET M = 0
320 PRINT T, X, , V
330 IF (T - 30) > = 0 THEN 570
340 REM * * * * * * * * * * * * * CALCULATE SLOPES
350 LET M = M + 1
360 LET X1 = X
370 LET V1 = V
380 LET A1 = C1*V1 + C2*X1
390 LET X2 = X1 + V1*D/2
400 LET V2 = V1 + A1*D/2
410 LET A2 = C1*V2 + C2*X2
420 LET X3 = X1 + V2*D/2
430 LET V3 = V1 + A2*D/2
440 LET A3 = C1*V3 + C2*X3
450 LET X4 = X1 + V3*D
460 LET V4 = A1 + A3*D
470 LET A4 = C1*V4 + C2*X4
480 REM
490 REM * * * * * * * * * NEW VALUES OF X, V, A, AND T
500 REM
510 LET X = X + (V1 + 2*V2 + 2*V3 + V4)*D/6
520 LET V = V + (A1 + 2*A2 + 2*A3 + A4)*D/6
530 LET A = C1*V + C2*X
540 LET T = T + D
550 IF M = 10 THEN 310
560 GO TO 330
570 END
```

```
SPRING CONSTANT = 2.8 NEWTONS/METER

DAMPING COEFFICIENT = .4 NEWTONS/METER/SEC

TIME           X-DISPLACEMENT                VELOCITY
 0              30                            0
 1.             24.9499                      -10.1607
 2.             11.5866                      -15.9949
 3.            -4.7886                       -15.801
 4.            -18.2324                      -10.1108
 5.            -24.2815                      -1.33245
 6.            -21.3957                       7.23454
 7.            -11.2697                       12.6184
 8.             2.00713                       13.1865
 9.             13.5613                       9.11154
 10.            19.4955                       2.17251
 11.            18.173                       -4.9898
 12.            10.6138                      -9.86009
 13.           -6.50444E-2                   -10.9162
 14.           -9.89872                      -8.08357
 15.           -15.5225                      -2.64969
 16.           -15.2991                       3.28984
 17.           -9.75677                       7.62444
 18.           -1.23836                       8.9659
 19.            7.05587                       7.07686
 20.            12.2503                       2.86502
 21.            12.7719                      -2.02169
 22.            8.79869                      -5.82688
 23.            2.06266                      -7.30666
 24.           -4.87397                      -6.12401
 25.           -9.57636                      -2.89649
 26.           -10.5765                       1.09269
 27.           -7.81064                       4.39351
 28.           -2.5336                        5.9077
 29.            3.22067                       5.24474
 30.            7.40841                       2.80357
```

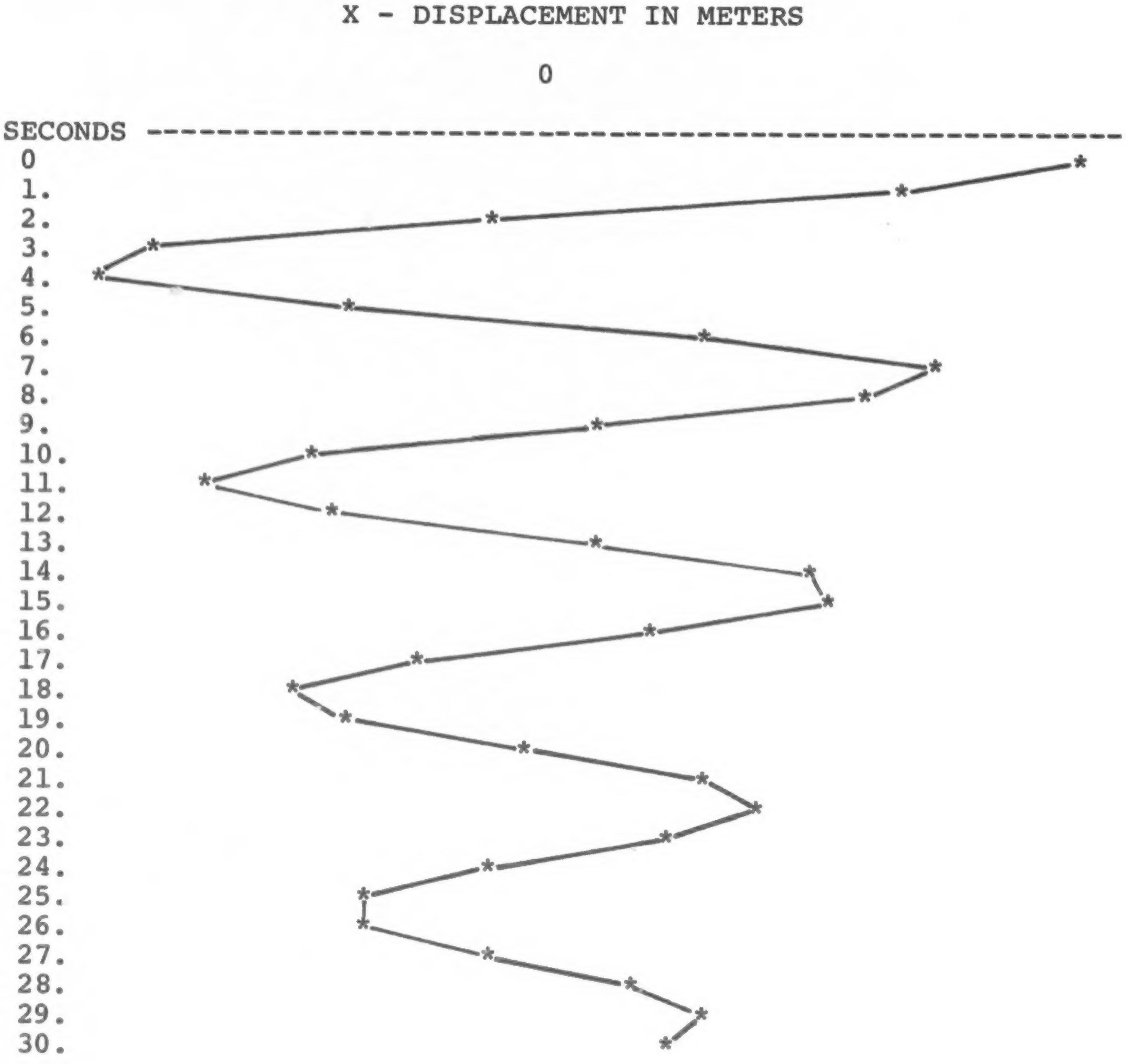
SPRING CONSTANT = 2.8 NEWTONS/METER

DAMPING COEFFICIENT = .4 NEWTONS/METER/SEC



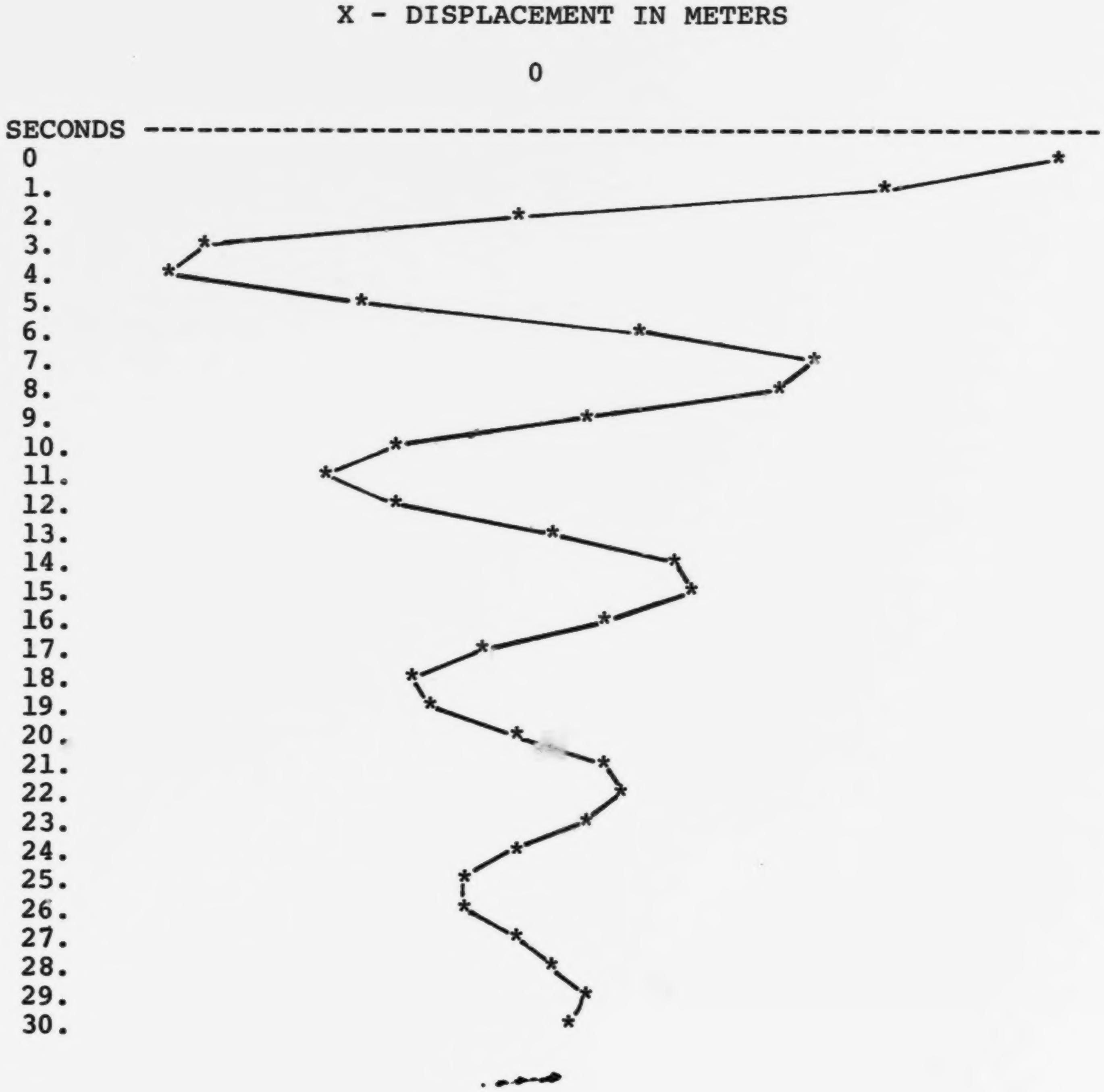
SPRING CONSTANT = 5.6 NEWTONS/METER

DAMPING COEFFICIENT = .4 NEWTONS/METER/SEC



SPRING CONSTANT = 5.6 NEWTONS/METER

DAMPING COEFFICIENT = .8 NEWTONS/METER/SEC



TEACHER'S GUIDE

In this presentation, the functional relationships have been developed in BASIC in order to facilitate the relation of the algorithms describing the physics directly to the computer program. For instance, in the first program initial conditions are defined in line numbers 210 through 240 of the program. The plotting of computer-generated output is directed by line numbers 280 and 340. The four slopes estimating velocity and acceleration are calculated in line numbers 390 through 500. Time is incremented in line 550. The statement SETDIGITS (3) in line 560 is used to set the size of the numbers to be printed by the computer, in this case, three significant digits. It is included solely for esthetic reasons.

Students should be encouraged to write a similar program and to explore the effects on the period of changing the spring constant, K, the mass, M, the initial displacement, X0, or the initial velocity, V0. New values of K, M, X0, and V0 can be entered in the data statement in line 160. The period can actually be determined from the plotted computer output, thus, allowing students to graph the period versus any of the above variables to determine an empirical relationship. (Recall that $T = 2\pi\sqrt{M/K}$.)

In the second program numerical output is provided. However, to enter the plot routine it is only necessary that line numbers 280 through 340 of the previous program must be added. Compare the numerical output of this program (numbers are printed to six digits) to the previous program, where the statement SETDIGITS (3) was used. Students should be encouraged to investigate the effect on period and amplitude of manipulating values of the spring constant, mass and damping coefficient.

Solution of the differential equation describing the motion of the damped spring-mass system can be found in most handbooks. For comparison of results, students should program both the numerical solution and the closed-form solution of this problem, which, for $T0 = 0$, is

$$X = e^{-BT/M}[X0 \cos\omega T + C \sin\omega T]$$

where $\omega = \sqrt{K/M}, \quad C = V0/\omega + (B/M)(X0/\omega)$

For the sake of realism, these equations have not been scaled to dimensionless variables. However, programming could have been simplified by scaling time to units of one period, $T = t/T$.

References

Halliday, D. and R. Resnick, Physics for Students of Science and Engineering, John Wiley & Sons, Inc. (1967).

Hamming, R. W., Numerical Methods for Scientists and Engineers, McGraw-Hill Book Company, Inc. (1962).

TWO EXPERIMENTS

Conservation of Momentum
and
Simple Harmonic Motion

David T. Grimsrud

Muhlenberg College
Allentown, Pennsylvania

INTRODUCTION

The two programs discussed here illustrate the use of the computer as a computational adjunct to an introductory physics demonstration or laboratory. The programs are intended to aid the student in analyzing the data acquired in the course of the experiment. The first program, MOMEN, was developed to give immediate answers for a lecture demonstration of the conservation of momentum. This demonstration will be part of a one-semester "Physics for Poets" course which does not have an associated laboratory. The students, for the most part, are second-year students who have no preparation in physics, but may have studied another laboratory science for one year.

Since this program has not been tested with any groups of students, we omit mention of any student materials, except to say that the students, by this juncture, will have been introduced to the computer and will understand simple programming concepts. The principle of conservation of momentum will have been discussed prior to the demonstration.

The second program, PEND1, was written for use in the general physics laboratory, in which simple harmonic motion is studied by recording the position of a pendulum bob (see Figure 1) as it moves through one cycle of its oscillation. This program has been used by approximately 60 first- and third-year science majors concurrently studying calculus. The method of calculating velocities and accelerations used in the program had been used previously in the course, although the students had no previous experience with the computer. The program was introduced in a written description given to the class a week prior to the experiment. The three-hour laboratory period was devoted to a summary explanation of the experiment and program, obtaining the data, keypunching these data, running the program, and plotting the results.

Inasmuch as these two programs have had little classroom testing, their student syllabi are not fully developed. However, a student manual for the simple harmonic oscillator is presented in the Appendix to this paper.

Conservation of Momentum

Experiments in the conservation of momentum in one dimension are quite common in the classroom and can take many forms: a linear air track may support two colliding gliders, two steel balls may be hung on bifilar suspensions, etc. In such an experiment some device must be provided to measure the position of the particles at equal intervals of time. An open-shuttered Polaroid camera and strobe illumination, or the spark-trace attachment found on most air tracks, would be appropriate. The positions indicated by these devices permit calculation of velocities and thence momenta; the momentum before collision is compared with the momentum after collision.

The program is divided into three parts: the velocity of each particle before and after collision is calculated; the momenta are determined; and the momentum of the system before the collision is compared with the momentum after the collision.

The input data required by the program are:

1. Masses of two particles, M(1) and M(2).
2. The time interval, DELT, between successive position measurements.
3. The number, L, of position measurements before collision (chosen such that the number after collision is also L).
4. The positions of the particles which form the three-dimensional array DISP(I, J, K).
    - a. DISP(I,1,1) I = 1, . . ., L give the positions of particle 1 before collision.
    - b. DISP(I,2,1) I = 1, . . ., L give the positions of particle 2 before collision.
    - c. DISP(I,1,2) I = 1, . . ., L give the positions of particle 1 after collision.
    - d. DISP(I,2,2) I = 1, . . ., L give the positions of particle 2 after collision.

The technique used to compute the velocity eliminates a common error the student might make if he were performing the calculation without direction. He is faced with the problem of finding the average interval between displacements separated by equal intervals of time. If we label these positions X(1), X(2), . . ., X(K), . . . and the intervals D(1) = X(2)-X(1); D(2) = X(3)-X(2); D(I) =

X(I+1)-X(I), his intuition will likely direct him to consider

$$\bar{D} = \frac{1}{n} \sum_{I=1}^{n} D(I)$$

as the average value of the interval. Substitution of the measured points X(1), . . ., X(n+1) into the above expression shows that it reduces to

$$\bar{D} = \frac{1}{n} \left(X(n+1)-X(1) \right)$$

which neglects most of the data obtained.

A more efficient technique (in the sense that each data point is used once to find the average interval) is effected if the position data are divided into two groups and paired to produce intervals roughly one-half the total interval. These half-intervals are then averaged, the time required to travel a half-interval is computed, and the average velocity is found.

Assume, as an example, that nine position measurements (L = 9) are made before the collision X(1), X(2), . . ., X(9). The pairings chosen to evaluate the average half-interval in this case would be X(9)-X(4), X(8)-X(3), X(7)-X(2), and X(6)-X(1). To choose the proper position index to subtract from X(9) we use integer division, dividing L by 2 to give the integer variable L2. In this example, L divided by 2 gives 9/2 which is equal to 4 in integer arithmetic. The sum of the four half-intervals for the first particle (X(9)-X(4)) + (X(8)-X(3)) + (X(7)-X(2)) + (X(6)-X(1)) is computed and stored as the variable named INTOT(1,1). The average of these four determinations is called AVINT(1,1) and is found by dividing by the real value of L2. Finally, since the value of AVINT(1,1) contains the distance traveled in five time intervals, we find the velocity of the first particle by dividing AVINT(1,1) by the number of time intervals multiplied by the time per interval.

It would have been preferable to perform data reduction in terms of standard deviations, thereby utilizing fully the data collected,* however, the sophistication of the students in this course does not admit of it, hence the above stratagem, which is clearly preferable to being apprehended in the act of taking useless data. The remainder of the computation proceeds very simply after velocities have been determined; momenta are calculated and values of momenta before and after collision are compared.

MOMEN and a sample of its output are shown on the following pages.

---

*"Data Reduction," Paul A. Smith, in Computer-Based Physics: An Anthology, R. Blum, et al, published by the Commission on College Physics (1969).

Program: MOMEN

```
C     THE 3-DIMENSIONAL ARRAY DISP(I,J,K) CONTAINS INFORMATION ABOUT THE
C     DISPLACEMENTS OF THE TWO MASSES INDEXED BY J.  K IS ONE FOR DIS-
C     PLACEMENTS BEFORE COLLISION, TWO FOR THOSE AFTER.  MASSES ARE DE-
C     NOTED BY M(J).  L GIVES THE NUMBER OF APPROXIMATELY EQUALLY SPACED
C     DISPLACEMENTS SEPARATED IN TIME BY THE INTERVAL DELT
      REAL M(2), INTOT(2,2), MOM1, MOM2
      DIMENSION DISP(15,2,2), AVINT(2,2), VEL(2,2)
      READ (2,10) M, DELT, L
   10 FORMAT(2F5.1, F4.3, I2)
      READ (2, 12)(((DISP(I,J,K), I = 1,L), J = 1,2),K = 1,2)
   12 FORMAT(2F4.1)
C     USE INTEGER DIVISION TO DIVIDE THE INPUT DATA INTO TWO PARTS
      L2 = L/2
      RL2 = L2
C     FIX VALUES OF L AND L2 SINCE THEY WILL CHANGE IN THE DO LOOPS
      N = L
      N2 = L2
C     COMPUTE TOTAL HALF INTERVAL VALUES
      DO 100 J = 1,2
      DO 100 K = 1,2
      INTOT(J,K) = 0.0
      L = N
      L2 = N2
   90 INTOT(J,K) = INTOT(J,K) + DISP(L,J,K) - DISP(L2,J,K)
      IF(L2 - 1) 100,100,80
   80 L = L-1
      L2 = L2 - 1
      GO TO 90
  100 CONTINUE
C     COMPUTE AVERAGE VELOCITIES
      TIME = DELT*((N+1)/2)
      DO 200 J = 1,2
      DO 200 K = 1,2
      AVINT(J,K) = INTOT(J,K)/RL2
      VEL(J,K) = AVINT(J,K)/TIME
  200 CONTINUE
   14 FORMAT(' VELOCITY OF THE FIRST PARTICLE BEFORE COLLISION WAS',
     XF6.1,/5X,' WHILE THAT OF THE SECOND PARTICLE WAS',F6.1)
      WRITE(3,14) VEL(1,1), VEL(2,1)
   16 FORMAT(' VELOCITIES OF THE PARTICLES AFTER COLLISION WERE',
     XF6.1,' AND',F6.1' RESPECTIVELY.')
      WRITE (3,16) VEL(1,2), VEL(2,2)
      MOM1 = M(1)*VEL(1,1) + M(2)*VEL(2,1)
   18 FORMAT(' TOTAL MOMENTUM OF THE SYSTEM BEFORE COLLISION WAS',
     XF8.1)
      WRITE(3,18) MOM1
      MOM2 = M(1)*VEL(1,2) + M(2)*VEL(2,2)
   20 FORMAT(' TOTAL MOMENTUM OF THE SYSTEM AFTER COLLISION, ON THE
     XOTHER HAND, WAS ', F8.1)
      WRITE(3,20) MOM2
      DISCR = ((MOM1 - MOM2) / (MOM1 + MOM2)) * 100.
```

```
22 FORMAT(' THUS OUR SIMPLE EXPERIMENT HAS SHOWN THAT MOMENTUM IS CON
  XSERVED WITHIN ', F6.1)
   WRITE(3,22) DISCR
   WRITE(3,24)
24 FORMAT(' PERCENT. WHAT EFFECTS CONTRIBUTE TO THIS DISCREPANCY.')
   CALL EXIT
   END
```

Output:

```
VELOCITY OF THE FIRST PARTICLE BEFORE COLLISION WAS   9.9
      WHILE THAT OF THE SECOND PARTICLE WAS   0.0
VELOCITIES OF THE PARTICLES AFTER COLLISION WERE   3.3 AND  13.3 RESPECTIVELY.
TOTAL MOMENTUM OF THE SYSTEM BEFORE COLLISION WAS   997.7
TOTAL MOMENTUM OF THE SYSTEM AFTER COLLISION, ON THE OTHER HAND, WAS   1007.7
THUS OUR SIMPLE EXPERIMENT HAS SHOWN THAT MOMENTUM IS CONSERVED WITHIN   -0.4
PERCENT. WHAT EFFECTS CONTRIBUTE TO THIS DISCREPANCY.
```

Simple Harmonic Motion

This program was written for use in the general physics laboratory at Muhlenberg. Simple harmonic motion is studied by making a record of the position of a pendulum bob as the bob moves through one cycle of its oscillation. A sketch of the apparatus is shown in Figure 1. A spark-trace record of the position of the bob is left on sensitized paper placed beneath the pendulum on a curved support. The student measures positions from the paper, plots these results, computes velocities from displacements, plots these values, and finally computes and plots accelerations. The three graphs obtained are compared to the trigonometric functions which describe the system.

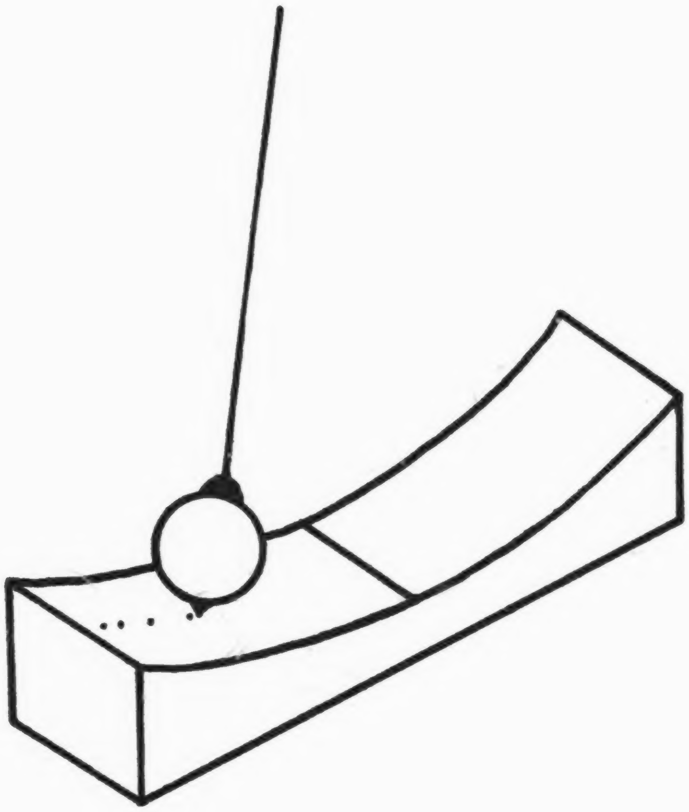


Figure 1

The input for the program is:

1. The time interval in seconds between successive sparks, DELT.
2. The total number of data points, J.
3. The ordered positions, in centimeters, of the J points.

We require that the position of the J points be measured from the midpoint of the motion (which was marked on the sensitized paper) so that the data have the approximate form R sin(ωt) where R =

amplitude of oscillation. A typical trace is sketched in Figure 2. Furthermore, the initial data point should be the last negative value of the position (1) of the bob before it swings positive. The J position coordinates are punched on cards using a free style input, i.e., the actual number (including its decimal point in the appropriate place) is punched, a space, another number, space, etc.*

The initial computation in the program is the angular frequency of the motion. The first zero of displacement from the vertical occurs (by design) between the first two points (1) and (2). This location is found by linearly interpolating between these points. Since the points then become positive, the next zero occurs just before the next subsequent negative position (3). Assuming that the midpoint of the motion has been correctly identified, the time between the two zeroes is one-half the period.

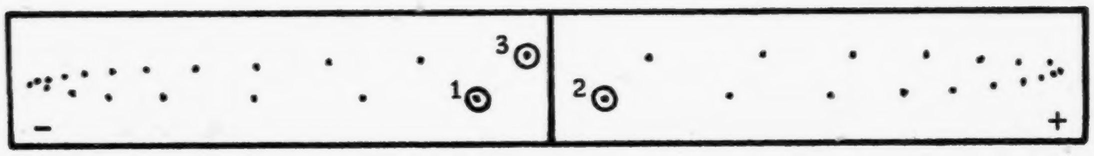


Figure 2

The phase angle and approximate amplitude R are then calculated, the latter simply being identified as the maximum of the absolute values of the displacement. Differences between successive positions are determined and, by multiplying by the reciprocal of the time interval, the average velocity in that time interval is found. Similarly, differences in velocity multiplied by the reciprocal of the time interval give average accelerations.

For purposes of plotting graphs the expected values of the displacement $R \sin(\omega t)$, velocity $R\omega \cos(\omega t)$, and acceleration $-R\omega^2 \sin(\omega t)$ are computed at times corresponding to the times for which the experimental values have been calculated.

The six columns of print-out are double spaced. In this way the velocities which approximate the instantaneous velocity at the midpoint of the interval between two displacements can be placed on lines which alternate with the displacements. In a similar fashion the accelerations are printed on lines which alternate with the velocities.

---

*Smith, Paul A., op. cit.

Program: PEND1

```
      DIMENSION X(90), V(90), DIFF(90), XSIN(90), DIF2(90), A(90),ASIN(9
     10), VCOS(90)
      READ (2,10) DELT, J
   10 FORMAT (F5.3, I2)
      IF(DELT) 50,50,12
   12 IF (J) 50,50,14
   14 CALL FREES (X,J)
C     CALCULATE THE ANGULAR FREQUENCY OF THE OSCILLATION
      ZERO = X(1)/(X(1) - X(2) )
      J2M = (J/2) - 4
      DO 60 I =  J2M,J
      IF (X(I)) 70,70,60
   60 CONTINUE
   70 ZERO1 = I - 2 + X(I-1)/(X(I-1) - X(I))
      T2 = (ZERO1 - ZERO ) * DELT
      OMEGA = 3.141593/T2
C     CALCULATE THE PHASE ANGLE OF THE MOTION
      PHI = ZERO * OMEGA * DELT
C     CALCULATE THE APPROXIMATE AMPLITUDE OF THE MOTION
      XMAX = X(1)
      DO 100 I = 2,J
      IF (ABS(X(I)) - XMAX) 100,100,90
   90 XMAX = ABS(X(I))
  100 CONTINUE
C     CALCULATE THE AVERAGE VELOCITY BETWEEN X(I) AND X(I+1)
      J1 = J - 1
      TDELT = 1.0 / DELT
      DO 200 I = 1, J1
      DIFF(I) = X(I+1) - X(I)
      V(I) = DIFF(I) * TDELT
  200 CONTINUE
C     CALCULATE THE AVERAGE ACCELERATION BETWEEN V(I) AND V(I+1)
      J2 = J - 2
      DO 300 I = 1, J2
      DIF2(I) = V(I+1) - V(I)
      A(I+1) = DIF2(I) * TDELT
  300 CONTINUE
C     COMPUTE THE EXPECTED VALUES OF DISPLACEMENT AND ACCELERATION
      ARG = OMEGA * DELT
      OMEG2 = OMEGA * OMEGA
      DO 400 I = 1,J
      RLI = I
      XSIN(I) = XMAX * SIN((RLI - 1.0) * ARG - PHI)
      ASIN(I) =-OMEG2 * XSIN(I)
  400 CONTINUE
C     COMPUTE THE EXPECTED VELOCITY AT THE MIDPOINT OF A DISPLACEMENT
C     INTERVAL
      AMPV = XMAX * OMEGA
      ARG2 = ARG * 0.5
      VCOS(1) = AMPV * COS(ARG2 - PHI)
      DO 500
      RLI = I
      VCOS(I) = AMPV * COS((RLI * ARG) - ARG2 - PHI)
  500 CONTINUE
      WRITE (3,16) OMEGA, XMAX
   16 FORMAT(' THE ANGULAR FREQUENCY AND AMPLITUDE HAVE VALUES' 2F10.2/)
      WRITE (3,18)
   18 FORMAT(' WHAT UNITS ARE ASSIGNED TO THE QUANTITIES LISTED ABOVE'/)
      WRITE (3,19) PHI
   19 FORMAT(' THE PHASE ANGLE HAS THE VALUE IN RADIAN MEASURE OF MINUS
     X ' F10.4 / )
      WRITE (3,20)
   20 FORMAT ( '  POSITION        RSIN(WT)        VELOCITY        RWCOS(WT)
     X  ACCELERATION    - RWWSIN(WT)')
      WRITE (3,21)
   21 FORMAT( '      CM              CM           CM/SEC          CM/SEC
     X CM/SEC/SEC       CM/SEC/SEC ' / )
      WRITE ( 3,22) X(1), XSIN(1), V(1), VCOS(1)
   22 FORMAT(F10.2, 6X, F9.2/ 31X, F9.1 , 6X, F9.1)
      WRITE ( 3,24) (X(I) , XSIN(I) , A(I)   , ASIN(I)  , V(I), VCOS(I),
     XI = 2, J1)
   24 FORMAT( F10.2, 6X, F9.2, 36X, F9.0, 6X, F9.0/ 31X, F9.1,6X,F9.1)
      WRITE (3,26) X(J),XSIN(J)
   26 FORMAT (F10.2,6X,F9.2)
   50 CALL EXIT
      END
```

Output:

```
THE ANGULAR FREQUENCY AND AMPLITUDE HAVE VALUES      3.00     33.05

WHAT UNITS ARE ASSIGNED TO THE QUANTITIES LISTED ABOVE

THE PHASE ANGLE HAS THE VALUE IN RADIAN MEASURE OF MINUS     0.0052

POSITION      RSIN(WT)    VELOCITY    RWCOS(WT)   ACCELERATION  - RWWSIN(WT)
  CM            CM        CM/SEC      CM/SEC      CM/SEC/SEC     CM/SEC/SEC

  -0.17        -0.17
                              97.1        99.2
   4.20         4.28                                   -59.          -38.
                              94.4        97.4
   8.45         8.67                                   -49.          -78.
                              92.2        93.9
  12.60        12.89                                   -14.         -116.
                              91.5        88.7
  16.72        16.88                                  -350.         -152.
                              75.7        81.8
  20.13        20.56                                   -54.         -186.
                              73.3        73.4
  23.43        23.86                                  -202.         -216.
                              64.2        63.7
  26.32        26.73                                  -276.         -241.
                              51.7        52.8
  28.65        29.11                                  -138.         -263.
                              45.5        41.0
  30.70        30.95                                  -355.         -280.
                              29.5        28.4
  32.03        32.23                                  -375.         -291.
                              12.6        15.3
  32.60        32.92                                  -153.         -297.
                               5.7         1.9
  32.86        33.01                                  -449.         -298.
                             -14.4       -11.5
  32.21        32.49                                  -212.         -294.
                             -24.0       -24.7
  31.13        31.38                                  -335.         -283.
```

29.37	29.69	-39.1	-37.5	-271.	-268.
27.06	27.46	-51.3	-49.5	-172.	-248
24.40	24.73	-59.1	-60.7	-256.	-223.
21.22	21.54	-70.6	-70.8	-212.	-194.
17.61	17.96	-80.2	-79.6	-59.	-162.
13.88	14.05	-82.8	-86.9	-182.	-127.
9.78	9.89	-91.1	-92.6	-64.	-89.
5.55	5.54	-94.0	-96.6	-118.	-50.
1.08	1.09	-99.3	-98.9	34.	-9.
-3.32	-3.37	-97.7	-99.3	74.	30.
-7.57	-7.77	-94.4	-97.9	9.	70.
-11.80	-12.04	-94.0	-94.8	138.	108.
-15.75	-16.08	-87.7	-89.9	108.	145.
-19.48	-19.83	-82.8	-83.3	212.	179.
-22.78	-23.22	-73.3	-75.3	162.	210.
-25.75	-26.18	-66.0	-65.8	177.	236.
-28.36	-28.66	-57.9	-55.2	301.	259.
-30.36	-30.62	-44.4	-43.5	311.	277.
-31.73	-32.02	-30.4	-31.0	162.	289.
-32.77	-32.83	-23.1	-18.0	375.	297.
-33.05	-33.04	-6.2	-4.6	553.	299.
-32.21	-32.64	18.6	8.7		

APPENDIX

Student Manual for Pendulum Experiment (Simple Harmonic Motion)

The apparatus used in this experiment [Figure 1] is a pendulum made of a pointed metal bob suspended from a long wire. A strip of sensitized paper is placed parallel to the trajectory of the bob on a curved metal track. At regular intervals of time a large potential difference is applied between the bob and the metal track causing a spark discharge. After one period of motion of the pendulum, the paper attached to the metal track contains a record of the position of the bob as a function of time. A typical paper record is shown below the apparatus sketch [Figure 2].

We must extract from the tape not only position vs. time information, but also the velocity and acceleration of the bob at equal intervals of time. This is a data reduction problem in which a large number of similar calculations must be performed. It is, therefore, an ideal problem to be handled by a computer.

We assume that a computer is still a "black box" to most of you, i.e., that you have had little or no experience working with one. It is then appropriate to give a brief description of the function of a computer.*

Imagine that you have employed a secretary who is efficient but not very intelligent. We can easily catalog the skills that she probably possesses. She can:

1. perform simple mathematical operations--addition, subtraction, multiplication, and division.
2. file information in definite locations and remember what these locations are.
3. follow carefully detailed instructions step-by-step.
4. compare two numbers and decide which is the greater.
5. type information as it accumulates.

These are not extreme standards to require in hiring a secretary. However, they represent the five functions performed by a computer. Why then are computers considered, for better or for worse, such an important part of our technological civilization? The answer is simple--speed. Because a computer can perform a calculation so much more rapidly than any other calculating device, it can complete any problem in a brief period of time that a human could do only in an unreasonable amount of time. This change of time scale permits us to contemplate operations with a computer which would be in the realm of fantasy without this tool.

---

*See, for example, Fortran for Physics, A. M. Bork, Addison-Wesley Publishing Co., Inc., Reading, Massachusetts (1967).

Muhlenberg installed an IBM 1130 computer system in the fall of 1968. Its justification in an educational institution depends on its usefulness as a computational tool, as a bookkeeping device, and as a teaching machine in Skinner's sense.

We wish to use the computer as a computational tool. To attack the problem of communicating with the computer, we must first decide what calculations we are to perform. Since we must make a graph of position, velocity, and acceleration and compare these with the theoretical values:

$$X = R \sin(\omega t - \phi)$$

$$V = V_m \cos(\omega t - \phi)$$

$$A = -A_m \sin(\omega t - \phi)$$

we see that we must calculate velocity, acceleration, and the quantities R, ω, and ϕ.

The program, i.e., the set of instructions which directs the computer to perform the calculations we need, is written in a language called FORTRAN (for FORmula TRANslation). The language was designed to conform closely to the actual structure of algebraic calculations. A copy of the program is attached to this supplement. This program is stored in the memory of the machine; to run the program you need only enter data (the positions of the bob) in the form of punched cards.

Perhaps the simplest way to understand both the computation procedure and the program is to look at the program and follow its statements through the calculation. Note that all statements prefixed by the letter C in the far left hand column are comments inserted into the program to clarify steps for the reader. They're ignored by the machine. We shall use them as "chapter headings."

C CALCULATE THE ANGULAR VELOCITY OF THE OSCILLATION

The angular velocity ω is equal to $2\pi/T$ where T is the period of oscillation. We assume the positions of the bob (measured from the midpoint of the motion) are entered into the machine consecutively beginning with the last negative value before the bob swings positive. We then find the time for which the pendulum bob had zero displacement. Subsequent positions are searched to find the next time the displacement was zero. The difference between these two times is one half the period; the angular velocity follows directly.

C CALCULATE THE PHASE ANGLE OF THE MOTION

The displacement X is given by the expression $R \sin(\omega t-\phi)$. When X is zero, the phase angle $\phi = \omega t$. The statement after the comment referred to above performs this calculation.

C CALCULATE THE APPROXIMATE AMPLITUDE OF THE MOTION

This step illustrates quite nicely two programming techniques, use of the IF statement and use of the DO loop.

The position data are denoted X(I) where I goes from one to the total number of data points. The technique we use to find the amplitude is simply to look at the absolute value of each X(I) and determine if it is larger than some other value of X(I) that we have assigned the value of XMAX. If it is larger we set the present value of X(I) equal to XMAX. XMAX is originally set equal to X(1).

Using the DO statement we let X(I) go from X(2) to X(J) (J is the total number of data points) by steps of one. Each value of X(I) is tested by the statement:

```
IF (ABS(X(I)) - XMAX) 100,100,90
```

This means:

1. Take the absolute value of X(I).
2. Subtract from this XMAX.
3. If the result is
   a. negative--go to statement 100.
   b. zero--go to statement 100.
   c. positive--go to statement 90.

Statement 90 changes XMAX from any previous value it had to a new value, viz. the absolute value of the present X(I). The program then moves to statement 100 and then back to the beginning of the DO loop to increment I by one until I = J.

C CALCULATE THE AVERAGE VELOCITY BETWEEN X(I) AND X(I+1)

You have made this calculation in two previous experiments.

```
V(I) = ( X(I+1) - X(I) )/ DELT
```

where DELT is the time interval between two successive points. We identify this velocity as the average velocity evaluated at the midpoint of the interval between X(I) and X(I+1). For this reason the velocity and position appear in the print-out spaced on alternate lines.

C CALCULATE THE AVERAGE ACCELERATION BETWEEN V(I) AND V(I+1)

This calculation is the same as that directly above

$$A(I+1) = (V(I+1) - V(I))/ DELT$$

This represents the average acceleration between the two velocities.

The program concludes by calculating

$$R \sin(\omega t - \phi)$$
$$R\omega \cos(\omega t - \phi)$$
$$-R\omega^2 \sin(\omega t - \phi)$$

for the values of time associated with the computations above.

Using the print-out, make three graphs: one of the displacement vs. time, the second of velocity vs. time, and the third of acceleration vs. time. Each should show experimental (or calculated) and theoretical results. Why does the agreement between theory and experiment become steadily worse as we move successively from displacement to velocity to acceleration?

RELATIVISTIC TWO-BODY COLLISIONS

Dale Winder

Colorado State University
Fort Collins, Colo.

INTRODUCTION

In the Physics Department at Colorado State University we have consciously endeavored to stimulate students to use the computer in their course work. In the introductory physics courses with calculus, we have fostered the generation of programs written by students and pertaining to the subject matter of the course on a voluntary basis. In the graduate-level Electromagnetic Theory course, problems were assigned to each student. The more experienced were given realistic problems, while novices were required to create function-generators. The program described here, ATOM, was written by a student in the third and last quarter of a course in general physics. It solves two types of problems: 1) the relativistic equations for the collision of two particles and 2) the transformation of coordinates between center-of-mass and laboratory frame. It is unnecessarily complicated by including three dimensions and the possibility of conversion between mass and kinetic energy.

All participants in the introductory course were treated as individuals and asked to create a program on their own (with consultation and help, of course). No manual was used, although a few introductory FORTRAN IV sessions were held for those who had no previous experience; in two one-hour sessions, I felt I could get enough across for the student to write simple programs involving DO-loops, functions, data and printing. The chief virtue of this method was that the novice students could choose a subject which interested them, analyze it, program an extensive (for them) calculation and make it work. The next step, if we carry on with this program, would be to have them tackle a nontrivial calculation, one that could not be computed in a finite time with a slide rule. The point in this effort is not in the quality of programs created, but in the involvement of the student in the use of a new, exciting, highly visible tool; one which many realize will be as common to them as a slide rule. There is an excitement and enthusiasm in solving problems of the relative difficulty and length represented by some of these problems.

STUDENT MANUAL

The program, ATOM, described here solves the conservation equations for the collision of two sub-atomic particles. The input data consists of the mass, velocity and direction of motion of two particles before their collision and of one particle after the collision. The program then computes the mass, velocity and direction of travel of the fourth particle. The total initial momentum of the two particles is then used to compute the transformation of variables to a frame of reference ("center-of-momentum" frame) where the total momentum of the system is zero.

The fundamental physical principle involved is the conservation of the relativistic momentum vector and the relativistic energy in the collision. For a single particle of rest mass m_0, speed v and velocity $\vec{v}$ these quantities are defined by the fundamental relativistic relations:

$$m = m_0(1 - v^2/c^2)^{-\frac{1}{2}} \quad \text{(relativistic mass)}$$

$$\vec{p} = m\vec{v} \quad \text{(relativistic momentum)}$$

$$E = mc^2 \quad \text{(relativistic energy)}$$

which imply

$$E^2 = p^2c^2 + m_0^2c^4$$

relating energy, momentum and rest mass (c = speed of light).

For a two-particle system the conservation equations are thus expressed as

$$\vec{p}_1 + \vec{p}_2 = \vec{p}_3 + \vec{p}_4$$

$$E_1 + E_2 = E_3 + E_4$$

assuming that two particles, 3 and 4, recoil from the collision, although they may not be of the same mass, individually or collectively, as the original colliding particles, 1 and 2. This is illustrated in Figure 1.

For ease of programming, it is necessary to "scale" the variables appropriately; we shall do this by means of the following substitutions:

$$V = v/c \ ; \quad M = m_0c^2 \ ; \quad \gamma = \sqrt{1 - V^2}$$

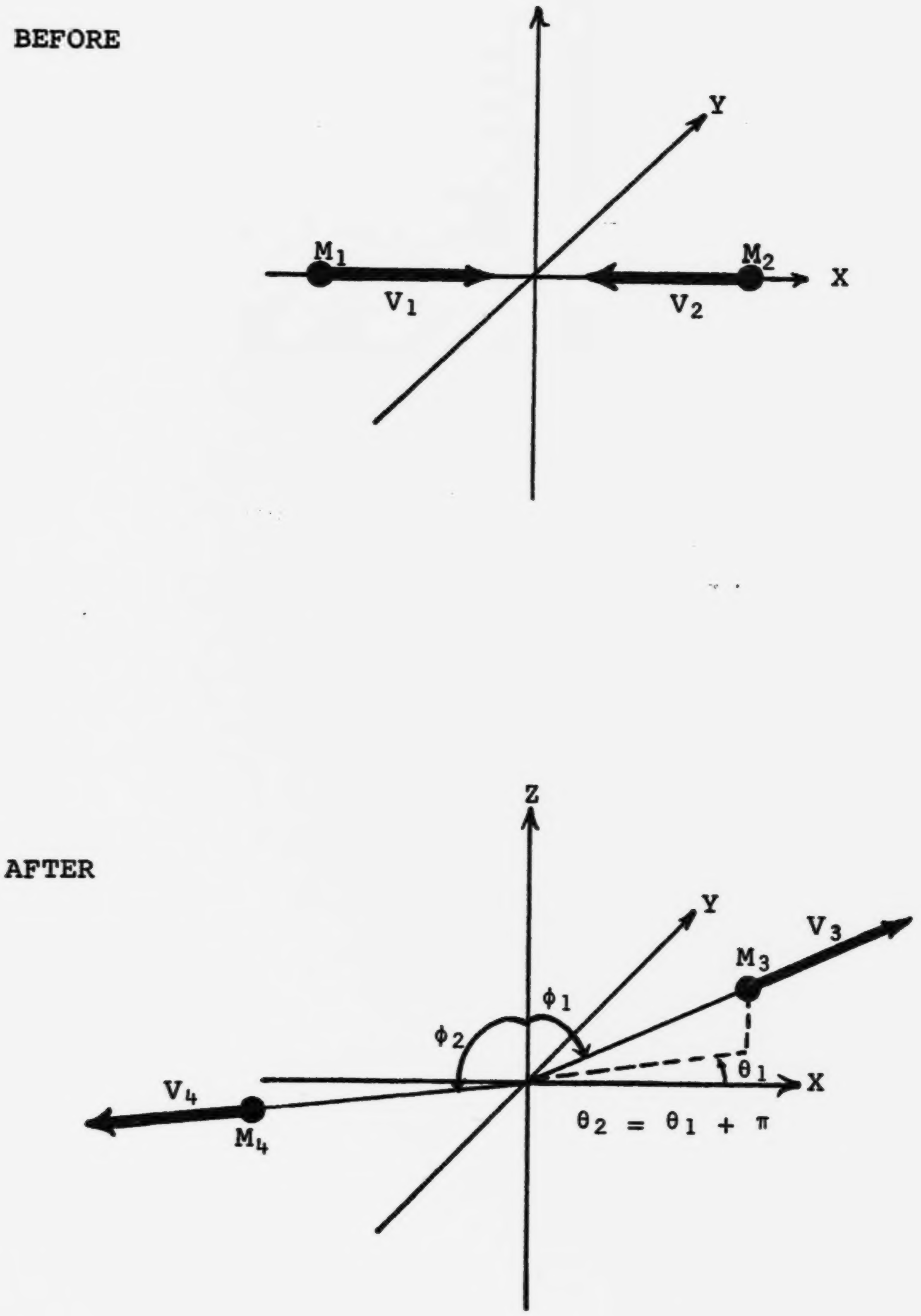

BEFORE
Z
Y
M_1
M_2
X
V_1
V_2
AFTER
Z
Y
V_3
M_3
ϕ_1
ϕ_2
θ_1
X
V_4
M_4
$\theta_2 = \theta_1 + \pi$

Figure 1

$$\therefore \quad \vec{P} = \vec{p}c = M\vec{V}/\gamma$$

$$E = M/\gamma \quad \text{and} \quad \vec{V} = \vec{P}/E$$

which also imply

$$E^2 = P^2 + M^2$$

The input data to ATOM are eight values $X_j = M_1, M_2, M_3, V_1, V_2, V_3, \phi_1$, and θ_1 and the index $I = 1$ for laboratory frame of reference, or $I = 2$ for center-of-momentum frame of reference. It is assumed that the direction of velocities and V determine the x-axis, and, although the z-axis could have been determined by requiring that it be in the plane of the x-axis and the recoil velocities, the program was purposely written in a general three-dimensional description. The direction angles, ϕ and θ were chosen as shown in Figure 1, and are assumed given in degrees.

Every quantity to be output by the program is stored as a two-dimensional array of form X(I,N) where I specifies the frame of reference. The program starts by assigning storage space to MASS, speed (VEL), magnitude of momentum (MOMENT), ENERGY, ϕ (PHI), θ (THETA), vector components of $\vec{V}_3$ ($VEL_{1,2,3}$), $\vec{V}_4$ ($VEL_{4,5,6}$), $\vec{P}_3$, $\vec{P}_4$, etc. In some arrays an extra space is set aside as a convenient "working space."

After specification of numerous FORMAT statements the program stores input data in the appropriate places (statement #41), and proceeds (#34) to compute momenta and energies for particles 1, 2, 3, and 4. After computing ϕ_1 and θ_1 in radians, along with the appropriate direction cosines, the components of $\vec{P}_3$ are calculated (#707), and $\vec{P}_4$ is found from the conservation of momentum. $\vec{V}_4$ is computed from $\vec{P}_4$ (#23), followed by V_4 (#24+1) and finally, the rest mass is found (#24+3). The computation, for the first frame of reference, is completed with the calculation of the direction angles ϕ_2, θ_2 of the fourth particle, in degrees.

Following this, the program transforms to the alternate frame of reference in ##702-800, essentially repeating the same computations as in the preceding paragraph, after which the stored values are output (##36-40) in their appropriate formats. A listing of ATOM and sample output is shown on the following pages.

```
       PROGRAM ATOM
       REAL MASS(2,5),VEL(2,4),MOMENT(2,5),ENERGY(2,5),PHI(2.2)
       REAL THETA(2,2),P(2,6),V(2,6),X(8),T(3),GAM(4),RUTH(2)
 1     FORMAT (I2,3(F9.3),3(F9.6),2(F9.3))
 2     FORMAT (1H0,*INPUT DATA IS FROM LABORATORY FRAME OF REFERENCE*)
 3     FORMAT (1H0,*INPUT DATA IS FROM CENTER OF MOMENTUM FRAME OF REFERE
      XNCE*)
 4     FORMAT (1H0,30X,*COLLISION AS SEEN FROM LABORATORY FRAME*,13X,*COL
      XLISION AS SEEN FROM CENTER OF MOMENTUM FRAME*)
 5     FORMAT (1H0,30X,4(*PARTICLE *,I1,2X,),3X,4(*PARTICLE *,I2,2X))
 6     FORMAT (1H ,10X,*MASS (MEV)    *,7X,4(F9.3,3X),*XX*,2X,4(F9.3,3X))
 7     FORMAT (1H ,10X,*VELOCITY (X 1/C)*,4X,4(F9.3,3X),*XX*,2X,4(F9.3,3X
      X))
 8     FORMAT (1H ,10X,*ENERGY (MEV) *,7X,4(F9.3,3X),*XX*,2X,4(F9.3,3X))
 9     FORMAT (1H ,10X,*MOMENTUM (MEV/C)*,4X,4(F9.3,3X),*XX*,2X,4(F9.3,3X
      X))
 10    FORMAT (1H0,10X,*MOMENTUM*)
 11    FORMAT (1H ,14X,*COMPONENTS*)
 12    FORMAT (1H ,19X,*X*,10X,4(F9.3,3X),*XX*,2X,4(F9.3,3X))
 13    FORMAT (1H ,19X,*Y*,10X,4(F9.3,3X),*XX*,2X,4(F9.3,3X))
 14    FORMAT (1H ,19X,*Z*,10X,4(F9.3,3X),*XX*,2X,4(F9.3,3X))
 15    FORMAT (1H0,10X,*VELOCITY*)
 17    FORMAT (1H ,19X,*X*,10X,4(F9.6,3X),*XX*,2X,4(F9.6,3X))
 18    FORMAT (1H ,19X,*Y*,10X,4(F9.6,3X),*XX*,2X,4(F9.6,3X))
 19    FORMAT (1H ,19X,*Z*,10X,4(F9.6,3X),*XX*,2X,4(F9.6,3X))
 20    FORMAT (1H0,10X,*DIRECTION OF*)
 50    FORMAT (1H ,10X,*TRAVEL*)
 51    FORMAT (1H ,13X,*PHI (DEGREES)*,4X,4(F9.3,3X),*XX*,2X,4(F9.3,3X))
 52    FORMAT (1H ,14X,*THETA (DEGREES) *,4(F9.3,3X),*XX*,2X,4(F9.3,3X))
 53    FORMAT (1H0,10X,*TOTAL MOMENTUM*,24X,F9.3,21X,*XX*,20X,F9.3)
 54    FORMAT (1H ,10X,*TOTAL ENERGY  *,24X,F9.3,21X,*XX*,20X,F9.3)
 56    FORMAT (1H0)
 58    FORMAT (1H0,20X,*VELOCITY OF CENTER OF MOMENTUM AS SEEN FROM LABOR
      XATORY  =  *,F18.7,* TIMES C .*)
       ZZ = 57.2957795
       A1 = 0.0
       A2 = 90.0
 57    A3 = 180.0
       A4 = 0.0
       READ(5,1) (I,(X(J),J=1,8))
       GO TO(39,40),I
 505     IF(X(4))861,862,862
 861   A4 = 180.0
 862   IF(X(5))41,860,860
 860   A3 = 0.0
 41    DO 21 J = 1,3
       MASS(I,J) = X(J)
 21    VEL(I,J) = X(J+3)
       PHI(I,1) = X(7)
       THETA(I,1) = X(8)
 34    DO 22 J = 1,3
       GAM(J) = SQRT(1.0 - VEL(I,J)**2)
       MOMENT(I,J) = (MASS(I,J) * VEL(I,J)) / GAM(J)
 22    ENERGY(I,J) =  MASS(I,J)        / GAM(J)
```

```
        MOMENT(I,5) = MOMENT(I,1) + MOMENT(I,2)
        ENERGY(I,5) = ENERGY(I,1) + ENERGY(I,2)
        ENERGY(I,4) = ENERGY(I,5) - ENERGY(I,3)
        PHIRAD = PHI(I,1) / ZZ
        THETAR = THETA(I,1) / ZZ
        T(1)  =             SIN(PHIRAD) * COS(THETAR)
        T(2)  =             SIN(PHIRAD) * SIN(THETAR)
        T(3)      =         COS(PHIRAD)
        DO 707 J = 1,3
        V(I,J) = VEL(I,3)*T(J)
707     P(I,J) = MOMENT(I,3)*T(J)
        P(I,4) = MOMENT(I,5) - P(I,1)
        P(I,5) =  - P(I,2)
        P(I,6) =  - P(I,3)
        M = 4
        DO 23 J = 4,6
23      V(I,J) = P(I,J)         / ENERGY(I,M)
        SUM1 = 0.0
        SUM2 = 0.0
        DO 24 J = 4,6
        SUM1 = SUM1 + V(I,J)**2
24      SUM2 = SUM2 + P(I,J)**2
        VEL(I,4) = SQRT(SUM1)
        MOMENT(I,4) = SQRT(SUM2)
        MASS(I,4)=SQRT(ENERGY(I,4)**2-MOMENT(I,4)**2)
        X = ACOS(V(I,6)/VEL(I,4))
        PHI(I,2) = X * ZZ
        XX = ATAN(V(I,5)/V(I,4))
        THETA(I,2) = XX * ZZ
        IF(V(I,5)) 701,702,702
701     THETA(I,2) = 360.0 - THETA(I,2)
702     IF(I.EQ.2) GO TO 26
        II = 2
        DO 704 J=1,2
704     RUTH(J)= MASS(I,J)/GAM(J)
        VCM=(RUTH(1)*VEL(I,1)+RUTH(2)*VEL(I,2))/(RUTH(1)+RUTH(2))
        GO TO 27
26      II = 1
        VCM=  -VEL(I,2)
27      DO 28 J=1,4
28      MASS(II,J) = MASS(I,J)
        DO 29 J=1,2
29      VEL(II,J)=(VEL(I,J)-VCM)/(1.0-VCM*VEL(I,J))
        GAMMA=SQRT(1.0-VCM**2)
        DO 31 J=1,4,3
        V7=1.0- VCM*V(I,J)
        V(II,J) =  (V(I,J)-VCM)/V7
        K1=J+1
        K2=J+2
        DO 31  L=   K1,K2
31      V(II,L) = (V(I,L) * GAMMA) / V7
        SUM1 = 0.0
        SUM2 = 0.0
        DO 32  J=1,3
```

```
      SUM1 = SUM1 + V(II,J)**2
      K1 = J+3
32    SUM2 = SUM2 + V(II,K1)**2
      VEL(II,3) = SQRT(SUM1)
      VEL(II,4) = SQRT(SUM2)
      DO 33 J=1,4
      GAM(J) = SQRT(1.0 - VEL(II,J)**2)
      ENERGY(II,J) = MASS(II,J)  /  GAM(J)
33    MOMENT(II,J)  =    (MASS(II,J)   *   VEL(II,J) )   /    GAM(J)
      MOMENT(II,5) = MOMENT(II,1) + MOMENT(II,2)
      ENERGY(II,5) = ENERGY(II,1) + ENERGY(II,2)
      M=3
      DO 25 J=1,6
      V7 =SQRT(1.0-V(II,J)**2)
      IF(J.EQ.4) M=4
25    P(II,J) = (MASS(II,M) * V(II,J)) / V7
      L7 = 3
      K = 3
      N = 2
      N4 = 1
      DO 36 J=1,2
      IF(J.EQ.1) GO TO 727
      L7 = 4
      K = 6
      N = 5
      N4 = 4
727   X = ACOS(V(II,K)  / VEL(II,L7))
      PHI(II,J) = X * ZZ
      XX = ATAN(V(II,N)  /  V(II,N4) )
      THETA(II,J) = XX*ZZ
      IF(V(II,N)) 800,36,36
800   THETA(II,J) = 360.0 - THETA(II,J)
36    CONTINUE
      WRITE(6,4)
      WRITE(6,5)  ((I,I=1,4),(J,J=1,4))
      WRITE(6,6)  ((MASS(I,J),J=1,4),I=1,2)
      WRITE(6,7)   ((VEL(I,J),J=1,4),I=1,2)
      WRITE(6,8)   ((ENERGY(I,J),J=1,4),I=1,2)
      WRITE(6,9)   ((MOMENT(I,J),J=1,4),I=1,2)
      WRITE(6,10)
      WRITE(6,11)
      WRITE(6,12) (((MOMENT(J,I),I=1,2),P(J,1),P(J,4)),J=1,2)
      WRITE(6,13)  ((A1,A1,P(J,2),P(J,5),J=1,2))
      WRITE(6,14)  ((A1,A1,P(J,3),P(J,6),J=1,2))
      WRITE(6,15)
      WRITE(6,11)
      WRITE(6,17) (((VEL(J,I),I=1,2),V(J,1),V(J,4)),J=1,2)
      WRITE(6 18)  ((A1,A1,V(J,2),V(J,5),J=1,2))
      WRITE(6,19)  ((A1,A1,V(J,3),V(J,6),J=1,2))
      WRITE(6,20)
      WRITE(6,50)
      WRITE(6,51)   ((A4,A3, PHI(J,1),PHI(J,2),J=1,2))
      WRITE(6,52)   ((A2,A2, THETA(J,1),THETA(J,2),J=1,2))
      WRITE(6,53)    ((MOMENT(J,5),J=1,2))
      WRITE(6,54)    ((ENERGY(J,5),J=1,2))
      WRITE(6,58)    ( VCM )
      DO 55 J=1,5
55    WRITE(6,56)
      GO TO 57
39    WRITE(6,2)
      GO TO 505
40    WRITE(6,3)
      GO TO 505
      END
```

```
INPUT DATA IS FROM LABORATORY FRAME OF REFERENCE

                              COLLISION AS SEEN FROM LABORATORY FRAME             COLLISION AS SEEN FROM CENTER OF MOMENTUM FRAME

                              PARTICLE 1  PARTICLE 2  PARTICLE 3  PARTICLE 4     PARTICLE  1  PARTICLE  2  PARTICLE  3  PARTICLE  4
          MASS (MEV)            939.500     939.500     939.500    1198.491   XX     939.500     939.500     939.500    1198.491
          VELOCITY (X 1/C)         .650       -.800        .600        .677   XX        .734       -.734        .681        .589
          ENERGY (MEV)         1236.291    1565.833    1174.375    1627.750   XX    1382.953    1382.953    1282.855    1483.050
          MOMENTUM (MEV/C)      803.589   -1252.667     704.625    1101.448   XX    1014.839   -1014.839     873.532     873.532

          MOMENTUM
              COMPONENTS
                   X            803.589   -1252.667     573.422   -1022.500   XX    1014.839   -1014.839     707.336    -730.161
                   Y              0.         0.         208.709    -208.709   XX      0.          0.         154.912    -170.358
                   Z              0.         0.         352.312    -352.312   XX      0.          0.         268.334    -293.103

          VELOCITY
              COMPONENTS
                   X            .650000    -.800000     .488279    -.628168   XX     .733820    -.733820     .601475    -.520282
                   Y           0.         0.            .177719    -.128219   XX   0.          0.            .162691    -.140729
                   Z           0.         0.            .300000    -.216441   XX   0.          0.            .274632    -.237559

          DIRECTION OF
          TRAVEL
             PHI (DEGREES)        0.       180.000      60.000     108.655   XX      0.         180.000      66.214     113.786
              THETA (DEGREES)    90.000      90.000      20.000     348.463   XX      90.000      90.000      15.136     344.864

          TOTAL MOMENTUM                         -449.077                     XX                         .000
          TOTAL ENERGY                           2802.125                     XX                     2765.905

                    VELOCITY OF CENTER OF MOMENTUM AS SEEN FROM LABORATORY  =           -.1602632 TIMES C .
```

```
INPUT DATA IS FROM CENTER OF MOMENTUM FRAME OF REFERENCE

                    COLLISION AS SEEN FROM LABORATORY FRAME          COLLISION AS SEEN FROM CENTER OF MOMENTUM FRAME

                    PARTICLE 1  PARTICLE 2  PARTICLE 3  PARTICLE 4      PARTICLE  1  PARTICLE  2  PARTICLE  3  PARTICLE  4
MASS (MEV)            939.500     939.500     939.500    1826.060   XX    939.500      939.500      939.500      1826.060
VELOCITY (X 1/C)        0.         -.976        .616        .891   XX       .800        -.800         .600          .360
ENERGY (MEV)          939.500    4279.944    1192.729    4026.716   XX   1565.833     1565.833     1174.375      1957.292
MOMENTUM (MEV/C)        0.      -4175.556     734.807    3588.864   XX   1252.667    -1252.667      704.625       704.625

MOMENTUM
    COMPONENTS
         X              0.      -4175.556    -559.311   -3479.023   XX   1252.667    -1252.667      573.422      -573.422
         Y              0.          0.        166.974     -94.774   XX     0.           0.          208.709      -208.709
         Z              0.          0.        290.474    -160.384   XX     0.           0.          352.312      -352.312

VELOCITY
    COMPONENTS
         X          0.          -.975610     -.511541    -.885443   XX    .800000     -.800000      .488279      -.292967
         Y          0.          0.            .174984    -.051831   XX   0.           0.            .177719      -.106631
         Z          0.          0.            .295384    -.087494   XX   0.           0.            .300000      -.180000

DIRECTION OF
TRAVEL
   PHI (DEGREES)        0.       180.000      61.350      95.634   XX     0.          180.000       60.000       120.000
    THETA (DEGREES)   90.000      90.000     -18.884     356.650   XX   90.000         90.000       20.000       340.000

TOTAL MOMENTUM                       -4175.556                      XX                     0.
TOTAL ENERGY                          5219.444                      XX                  3131.667

        VELOCITY OF CENTER OF MOMENTUM AS SEEN FROM LABORATORY  =          .8000000 TIMES C .
```

TEACHER'S GUIDE

Available for student use were a CDC 6400 and an IBM 1401, neither one in a time-sharing mode; the language was FORTRAN. Job cards were issued upon request by the instructor at the beginning of the quarter, each good for one ten-second, ten-page run on the 6400 at a cost of $1. Students prepared, submitted and debugged programs at their convenience. A "pep talk" on the importance and relevance of computer calculations in science and engineering and a selection of possible problems helps to involve the students and to assure participation.

This program, ATOM, has run well in numerous tests, although it is not safeguarded against zeros in denominators or imaginary roots which could occur, because input is so free. In most cases, we have restricted ourselves to elastic collisions in a plane (θ_1, $= 0$, $M_3 = M_1$ or $M_3 = M_2$).*

---

*Editor's Note: It might also prove of interest to compare output from this program with that from the nonrelativistic program MOMEN in D. T. Grimsrud's "Two Experiments" in Computer-Based Physics: An Anthology, R. Blum, et al, published by the Commission on College Physics (1969). Since conservation computations are performed in terms of momentum and energy, it should be possible to modify ATOM to include Compton scattering as a special case for which, say, $V_1 = 1 = V_3$. A logical branch in the case of input $V_1 = 1$ would then compute $E_1 = M_1 = h\nu = hc/\lambda$ and $P_1 = E_1$ for the photon, omitting such undefined computations as #34-#22, etc.

A SIMULATED ACCELERATOR LABORATORY

R. C. Mikkelson

Macalester College
St. Paul, Minnesota

INTRODUCTION

A discussion of high energy particle physics is an integral part of most modern physics courses. Laboratory exercises using bubble chamber photographs are commonly employed to amplify the experimental side of these studies while questions of conservation laws and allowed reactions are covered in class and problem assignments. It is often very difficult for a student to capture the true flavor of this field from classroom treatments alone. Can a student be led to discover, or at least to test, the conservation laws as an active research participant? Is there a way for him to build the particle spectrum as a result of his own ideas and imagination?

The computer-based laboratory exercise presented here is a first attempt to answer these questions. In this exercise the student is asked to imagine himself as a theoretical physicist who suggests experiments to a large accelerator laboratory. The general idea is to allow the student to predict high energy particle reactions and to test his predictions by "experiments" carried out in the computer-simulated accelerator laboratory.

This laboratory exercise has been used during one term in a pre-calculus course for freshmen, mostly prospective physics and mathematics majors, called Introductory Modern Physics.* Since the classroom discussion of the concepts involved in the laboratory exercise did not come until the last week in the term, these students were really given an opportunity to experience life as it confronts the working physicist. Some took the exercise as a challenge and only went to the literature after exhausting their ingenuity and understanding. Others began by reading about the field and then proceeded to test and verify their understanding. Either approach seemed rewarding and succeeded in getting the students to study the field with real enthusiasm. The only computer-related skill required of the students for this exercise was card punching. As used in this course the computer experiments by the students were carried out during a six week period, so a fast turn-around time was not essential. The computer used by the class was an IBM 1130 operated in a hands-on, open shop where the students could either submit jobs for batch processing or operate the computer themselves.

A revised version of the "Student Manual" that was used by 38 students in the fall of 1968 is presented in the next section which will both illustrate the flavor of the exercise and give the reader an introduction from the user's point of view. The "Teacher's Guide" which follows it contains more detailed information about the simulation program and its use by students and concludes with the data storage techniques used with the IBM 1130 system, a listing of the particles and reactions contained in the simulation and complete program listings.

---

*Interested parties may write the author to request further information about this unique course.

STUDENT MANUAL

Introduction

One of the most interesting fields of contemporary physics is particle physics, sometimes called "high energy" physics, because the particles involved in the experiments have very high kinetic energies. The reactions take place in the GeV (10^9 eV) region and above, whereas chemical reactions involve energies on the order of one eV or so.

Now the question comes to mind, "How can those of us at a liberal arts college, without access to a high energy accelerator, get some firsthand knowledge of this field?" In our course on modern physics we attempt to attack this problem by including two experiments related to this field. One of these has become an old standard in the course, the analysis of some actual bubble chamber photographs.* A rather extensive write-up is available for that experiment and you should read that write-up before proceeding with this new experiment. It is not necessary for you to have actually worked on the bubble chamber photographs before attempting this new project; but, it might make things clearer if you have.

The new experiment, the subject of this write-up, requires that you think of yourself as a theoretical "high energy" physicist. You are working at one of the large accelerator laboratories where you help plan new experimental projects and try to make sense out of the experimental results. This allows you to ponder the results of experiments, to ask the questions that lead to new experiments and, hopefully, to discover the answers to your questions.

You might well ask, "What facilities do I have at my command?" The answer to that is, "An entire accelerator laboratory--accelerator, hydrogen bubble chamber, technical staff, and a crew to scan the bubble chamber pictures." It will be up to you to decide on the type of particle you would like to fire into the bubble chamber. You will also choose the kinetic energy of this incident particle. And, finally, you get to instruct the scanners to search for up to three types of resulting particles by simply specifying the masses of these "predicted" particles. This is all accomplished by letting the computer simulate the entire operation of the accelerator laboratory.

By studying the types of reactions that can, and do, take place at very high energies, a list of "particles" was selected as being representative of those actually known to physicists. The initial configuration of the simulation allows only protons to be the target particles (a liquid hydrogen bubble chamber, remember?). Initially, the incident projectile particles must be π^- mesons, and as work progresses on the simulation other projectile particles will become available; but you have to discover them first using the π^- mesons. The work involved here is not just getting the data--the laboratory supporting personnel do that for you--but rather in trying to make some sense out of it and planning new experiments.

---

*See Welch Catalog, No. 2171--Bubble Chamber Photo Analysis Apparatus.

If you want to know more about the workings of an actual accelerator and how a $\pi^- + p$ reaction comes about, you should read the write-up for the bubble chamber photo analysis experiment. If you would like to know more about the particles you will "discover" during the course of your experiments, you should read Chapter 10 in Beiser's paperback Modern Physics; An Introductory Survey, Addison-Wesley, Reading, Mass. (1968), and from there head for the library to the latest books on nuclear physics and particle physics, and to Scientific American articles. A list of possible references may be found in the book by M. S. Livingston, Particle Physics, McGraw-Hill, New York, N.Y. (1968), pp. 222-224. If you want suggestions for further study, please ask your instructor.

As a theoretical physicist, what parameters do you think are necessary to specify a particle, that is, to differentiate it from another particle? Suppose you are told there exist two particles with rest mass of about 939 MeV/c^2, (c = speed of light) one with positive charge and a second with no charge. At the present time, physicists refer to these two particles as two "charge states of the nucleon." You already know these particles by the names proton and neutron. In addition we know of the anti-nucleon states, the anti-proton and anti-neutron. We thus use charge as one means for differentiating particles. As you read about the elementary particles, you will discover many other ways for differentiating between them, but keep in mind that particles of equal mass but different charge may very well be simply different charge states of the same "fundamental particle" like the "nucleon."

As you read Chapter 10 in Beiser's book, you will become aware that the lifetime of the particles is an important parameter. Reactions that are "allowed" to take place by a strong interaction will happen in the short time it takes for two particles to pass one another while moving at a relative speed of approximately c. If we take the diameter of a nucleon (proton or neutron) to be about 10^{-13}cm (one "fermi") this time comes out to be about 10^{-24} seconds. However, Mother Nature hasn't "allowed" everything to happen that fast. In nuclear physics some reactions are called "weak" interactions and they are found to take place about 10^{14} times slower than the strong interactions. The time associated with weak interactions is thus about 10^{-10} seconds. You will find both strong and weak interactions going on as indicated by the lifetimes.

In the procedure section you will be guided through a trial run. It is suggested that you have the computer do this run for you as the first step in using the "laboratory." The laboratory will be at your disposal during the remainder of the term so you may feel free to do experiments any time. It is intended that this exercise occupy at least one week's worth of laboratory time. Be sure to read the note at the end about "publishing" your results.

Experimental Procedure

For each experimental run you must supply the accelerator staff with the identity of your projectile particle (a π^- meson initially) by specifying its mass and charge (137 MeV/c^2 and $-e$ for π^-). Then you must specify the laboratory kinetic energy of this incident particle in MeV; 0-2500 MeV will cover the entire range of simulated reactions. Only reactions with threshold energies within 10% of your specified energy will be detected.

Finally, once you decide on a projectile particle, you must tell the picture scanners what you are looking for. That is, you must "predict" the masses of up to three particles you expect to have produced. The overwhelming number of pictures obtained from an experimental run prohibits a complete analysis, so the photo scanning process is programmed to reject any reactions that do not result in one of the three particle masses specified by you. The scanners can determine masses within ± 5%.

As a theoretician you are interested in the results of the experiments. When the scanning crew finds a particle with a mass specified by you, they will record the data and give you a rundown of the significant reactions that occur during the experiment. For example, suppose the $\pi^- + p$ collision produces two particles: (1) mass M_1 and charge Q_1, and (2) mass M_2 and charge Q_2. You will be told these masses and charges. Suppose further that particle (2) decayed after 10^{-8} seconds into two other particles that also appeared on the photograph. You will also be told the masses and charges for these decay products and the lifetime of the parent particle (2). Since these are threshold reactions the reaction products are assumed stationary in center-of-mass coordinates.

In order to get down to earth let us consider the $\pi^- + p$ reaction as a trial run. Do you suppose there exists a particle with half the nucleon mass? To be specific, we want to find out whether the reaction $\pi^- + p \rightarrow N + M(500)$ exists. Here we have written N to stand for a nucleon (regardless of its charge state; i.e., either a neutron or proton) and M(500) stands for the undiscovered particle with rest mass around 500 MeV/c^2. The increase in mass-energy involved in producing the 500 MeV/c^2 particle is the difference between the new mass and the pion mass, 500-137 = 363 MeV/c^2. But we also know that some of the pion kinetic energy is effectively "lost" as center-of-mass energy and is not available for appearance as rest mass. Using the conservation of momentum and total energy we can calculate the threshold kinetic energy E_{th} required for the pion to be able to produce this new particle. The result of this derivation is

$$E_{th}\text{(threshold kinetic energy)} = QM/2m_2 \qquad \text{(Eq. 1)}$$

where: Q = (initial total rest mass - final total rest mass)c^2

M = (total final plus initial rest masses)

m_2= (rest mass of target particle)

To be specific, $Q = (939 + 137 - 939 - 500)$ MeV = 363 MeV

$M = (939 + 500 + 939 + 137)$ MeV/c^2 = 2515 MeV/c^2

m_2= 939 MeV/c^2

Carry out the arithmetic and you predict a threshold kinetic energy of about 485 MeV as required for the incident pion. Let's pick 500 MeV as a round number recalling that our scanners give us a 5% tolerance on mass determinations.

The FORTRAN READ statement for this program is the following:

```
    READ(2,201)  M1, IQ1, KE, MOUT(1), MOUT(2), MOUT(3)
201 FORMAT(6I5)
```

For input data we specify, as integers, the incident particle mass, M1, its charge in units of e, IQ1, its kinetic energy, and then up to three mass values MOUT( ) we want the scanners to look for. If we want the scanners to look for the new 500 MeV particle, a nucleon and a pion, the data card would take the following form:

```
  137   -1  500  500  939  137
```

In addition we need some XEQ cards and cards to define some storage areas. The cards required to call the program and to specify our data are as follows:

```
Columns:        5   10   15   20   25   30   35   40

Card 1:     // JOB
Card 2:     // XEQ PHYHE   01
Card 3:     *FILES(10,PARTS),(1,REAC1),(2,REAC2)
Card 4:       137   -1  500  500  939  137
Card 5:     blank
```

The computer will digest these five cards and print the following messages:

```
INITIAL EXPERIMENTAL CONDITIONS
INCIDENT PARTICLE MASS=  137 MEV/C2   WITH CHARGE= -1

KINETIC ENERGY OF PARTICLE BEAM =   500 MEV

SCAN IS FOR MASSES   500,  939, AND    137 MEV/C2

** PI MESONS ARE TOO NUMEROUS FOR A SCANNING REPORT
   THE SCANNERS ARE IGNORING YOUR REQUEST

SCAN HAS DETECTED NO PARTICLES WHICH SATISFY YOUR CONDITIONS.
```

As you can read, the computer will verify that it understands your specified initial conditions and then repeats your request for the scanning masses. In this case the scanners will not look for pi mesons (MOUT(3)) because they are simply too numerous, and finally we are told that no reaction of the kind we sought has been found.

Now here is the first opportunity to make an educated guess. Suppose you have a theory that some sort of particle exists which is really a collection (molecule?) of five bound pions. You might guess its mass to be 5×137=685 MeV/c^2. If you use this mass to calculate the laboratory threshold kinetic energy (Eq. 1) you predict about 790 MeV as the required energy for the incident pion. We can now do a new experiment simply by replacing Card 4 by a new card.

```
Card 4:        137   -1  800  500  939  685
```

where the incident pions have a kinetic energy of 800 MeV and the scan will be made for particles of mass 500, 939, and 685 MeV/c^2. The program yields the output shown on the next page (Figure 1).

Surprise! There is a particle with about half the mass of a nucleon. But, the most surprising part seems to be that it is not produced as simply as we had supposed. We had to supply more energy, apparently enough so that a new heavier particle could be produced to accompany our 496 MeV particle. Why didn't the neutral 496 MeV particle appear in the first experiment when there was certainly enough energy? Is there a conservation law acting here that prevents the production of the neutral 496 MeV particle in conjunction with, say, a neutron? Are there also charged particles with the masses 1115 MeV/c^2 and 496 MeV/c^2? We should note that no 685 MeV/c^2 particle was found. Does that mean it doesn't exist or that we simply don't know how to produce it?

If you look at the decays of these two new particles you find that the 1115 MeV particle decays in 2×10^{-10} seconds into either a proton and pi-minus meson or into a neutron and pi-zero meson. A weak interaction? What prevents a fast decay? Now look at the 496 MeV particle's decay habits. Nothing but puzzles! It seems to be a Jekyll-and-Hyde particle. Is it one particle with two different decay paths, or two particles with the same mass and

```
INITIAL EXPERIMENTAL CONDITIONS
INCIDENT PARTICLE MASS=  137 MEV/C2  WITH CHARGE= -1

KINETIC ENERGY OF PARTICLE BEAM =  800 MEV

SCAN IS FOR MASSES   500,  939, AND   685 MEV/C2

SUCCESS.   RESULTS FOLLOW.

PARTICLE (1)   CG=  0   MASS= 1115
PARTICLE (2)   CG=  0   MASS=  496

DECAY PRODUCTS OF PARTICLE (1) FOLLOW.
     DECAY TIME =  0.20E-09  CG =  1  MASS =   939
                             CG = -1  MASS =   137

     DECAY TIME =  0.20E-09  CG =  0  MASS =   939
                             CG =  0  MASS =   137


DECAY PRODUCTS OF PARTICLE (2) FOLLOW.
     DECAY TIME =  0.70E-10  CG =  1  MASS =   137
                             CG = -1  MASS =   137

     DECAY TIME =  0.70E-10  CG =  0  MASS =   137
                             CG =  0  MASS =   137

     DECAY TIME =  0.40E-07  CG =  1  MASS =   137
                             CG = -1  MASS =   137
                             CG =  0  MASS =   137
     DECAY TIME =  0.40E-07  CG =  0  MASS =   137
                             CG =  0  MASS =   137
                             CG =  0  MASS =   137
```

Figure 1

charge? It either decays into two pions with a lifetime of 0.7×10^{-10} seconds or it decays into three pions with a longer lifetime of 4×10^{-8} seconds. What is going on here?

Conclusion

Now you are on your own. It is hoped that you will come to appreciate the methods, joys, and tribulations of the theoretical physicist and that you will learn about the strange world of high energy physics, while having fun doing it. You should not sit in the computer room grinding out lots of numbers; instead, try to think up experiments that will provide you with answers to your questions. Keep track of your hypotheses, experiments, and miscellaneous thoughts in your lab notebook. The reactions available to you do not exhaust all of those possible in nature, but every effort has gone into providing a representative sample.

If you are lucky enough to discover new particles, you should "publish" your results on the bulletin board. Your instructors have started a "publication sheet" with their discoveries; you can add your results to it, unless someone else publishes first. However, verify ideas carefully before "publication," and beware the occupational disease of R.I.P. (Rushed Into Print).

To communicate your guesses or questions to other class members, simply write a note and stick it on the bulletin board. It is hoped that we may develop a small community of researchers, all working and sharing results in their common goal of understanding nature.

TEACHER'S GUIDE

Many simulations are designed for real time interaction between the user and the simulator. This acceleration laboratory simulation is not. Rather, it is intended that a student will try experimental runs primarily as tests for his ideas and will spend some period of time between these runs thinking about the results and planning new experiments. In order to encourage this and to discourage a shotgun approach the program presented here reads only one experimental specification per program execution, hence, it is readily usable with nearly all computer systems, time-shared or batch processed. In some cases there may be cost considerations, a desire for short turnaround times, or limitations on the number of computer runs allowed by any individual or class per day. In situations like these, being able to read more than one experimental specification per computer job could be easily achieved through a minor modification of the source program, such as relying upon an unsatisfied READ statement or a test for negative M1 to terminate the program.

Our students were encouraged to "publish" their discoveries by filling in a chart that was tacked on a bulletin board outside the lecture room. The chart included columns for particle charge, mass, lifetime, decay paths, and production technique. Additional space on the bulletin board was reserved for theoretical "papers" which consisted of short notes (by both students and instructor) asking questions about the discoveries and pointing out special properties. The students who worked hardest on the experiments quickly discovered many of the easy particles and the other students required reassurance that not everything had been discovered. One pair of students discovered that the K^- was available for use as an additional incident particle before this fact was revealed to the class by the instructor. Once the K^- became available many new reactions were possible and the late starters were able to get into the act.

It would be a mistake to expect students to arrive at a complete understanding of all the variations possible in this exercise without a fair amount of outside help. In addition to being encouraged to share their results and ideas with each other, our students were expected to use other references, the two primary ones being the book by Livingston, Particle Physics, McGraw-Hill, New York, N.Y. (1968) and the article by Chew, Gell-Man, and Rosenfeld, Scientific American (February 1964).*

---

*See Editor's Note on following page.

An opportunity for classroom discussion is essential for collecting loose ends. Coming after the experimentation, such a discussion can draw examples from the students' own "published" results and can point out the limitations in the simulation program. Furthermore, having prior information about the baryon spectrum did not seem to dampen enthusiasm among our students. Indeed, a theoretical prediction that a certain particle cannot be produced by either the $\pi^- + p$ or the $K^- + p$ reaction can be of more value than the publication of an accidental discovery of some new particle.

This simulation exercise was developed for use in the laboratory portion of an introductory course in modern physics. However, there do seem to be other possible uses for this program. For example, if a time-shared terminal could be made available in the classroom, such a simulation program might be very useful in a discussion session. Another possibility might be a problem set where the students could use the simulation in response to leading questions or as a check on their answers.

Editor's Note: To obtain the formula $E_{th} = QM/2m_2$ given in the Student Manual, recall that for an assembly of particles the quantity

$$(\text{total energy})^2 - (\text{total momentum})^2 c^2$$

is invariant with respect to Lorentz transformations and that energy is conserved in the collision. In the center-of-mass system the total momentum is, by definition, zero; and the energy before collision, E', equals E", the energy after collision. However, from the invariant, E' and E" are related to laboratory observations as follows:

$$E'^2 = (m_1 + m_2 + E_{th})^2 - p_1{}^2c^2 \ ; \quad E'' = M''$$

where M'' is the rest mass of the reaction products and p_1 the relativistic momentum of the projectile particle

$$p_1{}^2c^2 = (m_1 + E_{th})^2 - m_1{}^2 = 2m_1E_{th} + E_{th}{}^2$$

Making the appropriate substitutions in the equation $E'^2 = E''^2$ yields

$$M''^2 - (m_1 + m_2)^2 = 2E_{th}m_2$$

and factoring the left-hand side of the equation as the difference of two squares leads directly to the desired relationship.

This method is discussed, with examples, in Appendix A of Introduction to Elementary Particles, by W. S. C. Williams, Academic Press, New York, N.Y. (1961). The result can also be derived by straightforward but tedious calculations from conservation of energy and momentum. These can be found worked out in detail in The Atomic Nucleus, by R. D. Evans, McGraw-Hill, New York, N.Y. (1955), Chapter 12.

Many of the limitations contained in the present program are artificial and were included in order to reduce the development and programming time or to limit the student's freedom. For example:

1. The only particles available in the program are the baryons up to mass 1688 MeV/c^2, the π and K mesons, and the muons.

2. The particle manifold masses are combined. For example, both the proton and neutron appear as having a mass 939 MeV/c^2 and all the π-mesons appear with mass 137 MeV/c^2.

3. Only reactions having thresholds within 10% of the specified threshold energy are detected. This prevents picking a high energy and learning about all the reactions with lower thresholds in a single experiment.

4. No possibility has been included for reactions produced by secondary particles. Only $\pi^- + p$ and $K^- + p$ reactions and the decays of their resulting particles are included.

5. No information on the relative frequency of the various reactions and decays is included.

6. Electrons, gamma rays and neutrinos are ignored as being undetected.

A good case can be made for eliminating many of these limitations in future versions of the simulation, particularly if it is to be used by advanced students. It would not be difficult to include more members of the meson family or to increase the energy range and number of available reactions. The decay schemes could be greatly improved if electrons and gamma rays were included in the particle list.

The Program

The basic operation of the program is as follows:

1. Input data is read and a list (Table 1) of 33 particles is scanned to see if any of the given masses correspond to items on the list. Each item is indexed and the identifying indices of legitimate particle masses stored for later reference.

2. The program then scans a list (Table 2) of 24 possible $\pi^- + p$ or 32 possible $K^- + p$ reactions, depending on M1 and the threshold energy to determine which, if any, could give rise to the anticipated particles.

3. If an otherwise eligible reaction from Table 2 is found

not to contain the anticipated particles, then Table 1 is checked again to see if they might not be decay products of the reaction being examined. If successful, the program writes the result before returning to the next eligible reaction, if any, in Table 2.

As an illustration of how to read the particle table, Table 1 on the following page, we shall consider particle 21 which has a mass of 1405 MeV/c^2, charge 0, mean life 0.5×10^{-23} seconds. The 12 numbers following the lifetime are the decay paths recorded in four groups of three particles. It follows that particle 21 has three decay paths (zeros have no meaning) into the particle pairs (9,-1), (10,2), and (11,1). These integers refer to other particle numbers and may be looked up in the particle table. Negative particle numbers indicate the anti-particle. From the last column of Table 1 we can identify the particle as a Λ^{0*}; its three decay paths are (Σ^+,π^-), (Σ^0,π^0), and (Σ^-,π^+).

In Table 2 (page following Table 1) the reactions are listed in order of increasing threshold energy. Reaction products are given by particle numbers as found in the particle table. For example, consider reactions 13 and 14 having a threshold energy of 219 MeV. These produce particles (7,4,2) and (7,5,2), which may be identified as being $(n,K_1{}^0,\pi^0)$ and $(n,K_2{}^0,\pi^0)$.

This simulation program was developed for use with the IBM 1130 version 2 disk monitor system. In this system, the above particle table and reaction lists were stored as DATA FILES on the system disk, using the auxiliary program listed in Figure 2.

Some readers may not be familiar with the DEFINE FILE statement or with the Disk WRITE(NUM'J) XXX statement which appears as statement numbers 1 and 2 in this program. The DEFINE FILE statement gives information allowing the compiler to set up a data file on the disk. The WRITE(') statement is an instruction to "write on disk." Further details may be found in *Using the IBM 1130*, by A. Bork, Addison-Wesley, Reading, Mass. (1968).

The simulation program itself was also stored on the disk and could be called by the students using the short five-card deck listed in the Student Manual. This program was stored under the name PHYHE (Physics High Energy), hence the presence of that name on the program execute card (// XEQ). The third card of that deck is used to relate files numbered 10, 1, and 2 in the program READ(') statements to the names PARTS, REAC1, and REAC2 that were given to the particle table and reaction lists when stored on the disk. The fourth card contains the input data, and the last blank card simply insures that the card reader will read the fourth card if no jobs follow this one.

The flow chart on page 100 is a rough presentation of the simulation logic. It may be of some help to persons who attempt a detailed understanding of the program, but it is included here primarily to illustrate the gross program operation.

PART NUM	MASS	CHG	LIFE TIME	DECAY PRODUCT PARTICLE NUMBERS GROUPS OF THREE												PART SYM
1	137	1	0.20E-07	33	0	0	0	0	0	0	0	0	0	0	0	π
2	137	0	0.10E-15	0	0	0	0	0	0	0	0	0	0	0	0	
3	496	1	0.80E-08	1	2	2	1	1	-1	1	2	0	33	0	0	K
4	496	0	0.70E-10	1	-1	0	2	2	0	0	0	0	0	0	0	
5	496	0	0.40E-07	1	-1	2	2	2	2	0	0	0	0	0	0	
6	939	1	0.00E-00	0	0	0	0	0	0	0	0	0	0	0	0	N
7	939	0	0.00E-00	0	0	0	0	0	0	0	0	0	0	0	0	
8	1115	0	0.20E-09	6	-1	0	7	2	0	0	0	0	0	0	0	Λ
9	1193	1	0.60E-10	6	2	0	7	1	0	0	0	0	0	0	0	Σ
10	1193	0	0.10E-13	8	0	0	0	0	0	0	0	0	0	0	0	
11	1193	-1	0.10E-09	7	-1	0	0	0	0	0	0	0	0	0	0	
12	1236	2	0.10E-23	6	1	0	0	0	0	0	0	0	0	0	0	Δ
13	1236	1	0.10E-23	7	1	0	6	2	0	0	0	0	0	0	0	
14	1236	0	0.10E-23	6	-1	0	7	2	0	0	0	0	0	0	0	
15	1236	-1	0.10E-23	7	-1	0	0	0	0	0	0	0	0	0	0	
16	1318	0	0.10E-09	8	2	0	0	0	0	0	0	0	0	0	0	Ξ
17	1318	-1	0.10E-09	8	-1	0	0	0	0	0	0	0	0	0	0	
18	1382	1	0.50E-22	8	1	0	9	2	0	10	1	0	0	0	0	Σ*
19	1382	0	0.50E-22	8	2	0	9	-1	0	10	2	0	11	1	0	
20	1382	-1	0.50E-22	8	-1	0	10	-1	0	11	2	0	0	0	0	
21	1405	0	0.50E-23	9	-1	0	10	2	0	11	1	0	0	0	0	Λ*
22	1512	1	0.10E-23	6	2	0	7	1	0	0	0	0	0	0	0	N*
23	1512	0	0.10E-23	7	2	0	6	-1	0	0	0	0	0	0	0	
24	1520	0	0.20E-22	10	2	0	6	-3	0	7	-4	0	8	-2	0	Λ**
25	1530	0	0.60E-21	16	2	0	17	1	0	0	0	0	0	0	0	Ξ*
26	1530	-1	0.60E-21	17	2	0	16	-1	0	0	0	0	0	0	0	
27	1660	1	0.10E-21	6	-4	0	6	-5	0	9	2	0	21	1	0	Σ**
28	1660	0	0.10E-21	7	-4	0	7	-5	0	10	2	0	21	2	0	
29	1660	-1	0.10E-21	7	-3	0	11	2	0	21	-1	0	0	0	0	
30	1676	-1	0.10E-09	8	-3	0	17	2	0	16	-1	0	0	0	0	Ω
31	1688	1	0.10E-23	6	2	0	7	1	0	0	0	0	0	0	0	N**
32	1688	0	0.10E-23	7	2	0	6	-1	0	0	0	0	0	0	0	
33	106	1	0.20E-05	0	0	0	0	0	0	0	0	0	0	0	0	μ

Table 1. Particle List

$\pi^- + p$

REAC NUM	THRESH ENERGY	RESULTING PARTICLES		
1	268	14	0	0
2	765	8	4	0
3	765	8	5	0
4	831	22	-1	0
5	831	23	2	0
6	903	11	3	0
7	903	10	4	0
8	903	10	5	0
9	1011	8	4	2
10	1011	8	5	2
11	1157	31	-1	0
12	1157	32	2	0
13	1262	19	4	0
14	1262	19	5	0
15	1262	20	3	0
16	1308	21	4	0
17	1308	21	5	0
18	1369	7	3	-3
19	1369	7	4	-4
20	1369	7	5	-5
21	1369	6	-3	-4
22	1369	6	-3	-5
23	2225	17	3	-4
24	2225	17	3	-5

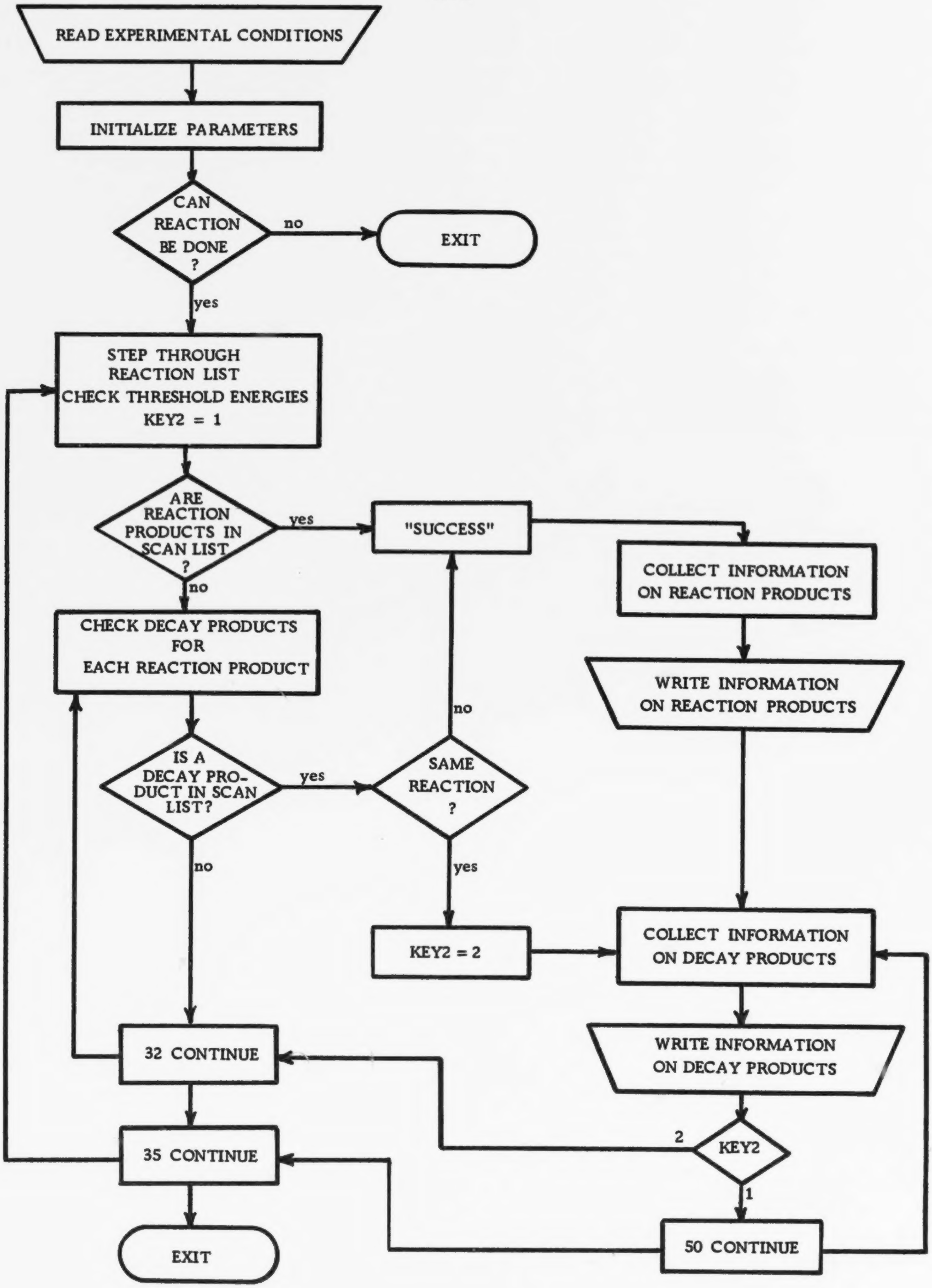
$K^- + p$

REAC NUM	THRESH ENERGY	RESULTING PARTICLES		
1	0	9	-1	0
2	0	10	2	0
3	0	11	1	0
4	0	7	4	0
5	0	7	5	0
6	132	18	-1	0
7	132	19	2	0
8	132	20	1	0
9	132	10	1	-1
10	132	8	1	-1
11	132	8	2	2
12	170	21	2	0
13	219	7	4	2
14	219	7	5	2
15	273	18	-1	2
16	273	19	2	2
17	273	20	1	2
18	366	24	2	0
19	623	27	-1	0
20	623	28	2	0
21	623	29	1	0
22	656	16	4	0
23	656	16	5	0
24	656	17	3	0
25	930	17	4	1
26	930	17	5	1
27	1089	25	4	0
28	1089	25	5	0
29	1089	26	3	0
30	1267	8	3	-3
31	2691	30	3	4
32	2691	30	3	5

Table 2. Reaction Lists

```
      DEFINE FILE 10(33,16,U,NEXT),1(24,4,U,NREC),2(32,4,U,NREC)
      DIMENSION NOS(3,4),IP(3)
      WRITE (3,300)
      DO 1 J=1,33
      READ (2,201)M,IQ,T,NOS
      WRITE (3,301) J,M,IQ,T,NOS
    1 WRITE (10'J)M,IQ,T,NOS
      MAXR = 24
      NUM = 1
      WRITE (3,303)
   11 DO 2 J=1,MAXR
      READ (2,202)KE,IP
      WRITE(3,302) J,KE,IP
    2 WRITE (NUM'J)KE,IP
      NUM = NUM+1
      GO TO (99,3,99),NUM
    3 MAXR = 32
      WRITE (3,303)
      GO TO 11
   99 CALL EXIT
  201 FORMAT (2I5,E5.0,12I5)
  202 FORMAT (4I5)
  300 FORMAT('1 PART MASS  CHG   LIFE'9X,'DECAY PRODUCT PARTICLE NUMBERS
     1'  / '      NUM              TIME'12X,'     GROUPS OF THREE'/  )
  301 FORMAT (' 'I4,I6,I5,E10.2,4(I6,2I4))
  302 FORMAT(I5,I9,5X,3I5)
  303 FORMAT('1 REAC  THRESH        RESULTING'/
     1       '   NUM  ENERGY        PARTICLES')
      END
```

Figure 2


READ EXPERIMENTAL CONDITIONS
INITIALIZE PARAMETERS
CAN REACTION BE DONE ?
no
EXIT
yes
STEP THROUGH REACTION LIST CHECK THRESHOLD ENERGIES KEY2 = 1
ARE REACTION PRODUCTS IN SCAN LIST ?
yes
"SUCCESS"
COLLECT INFORMATION ON REACTION PRODUCTS
no
CHECK DECAY PRODUCTS FOR EACH REACTION PRODUCT
WRITE INFORMATION ON REACTION PRODUCTS
no
IS A DECAY PRODUCT IN SCAN LIST?
yes
SAME REACTION ?
no
yes
KEY2 = 2
COLLECT INFORMATION ON DECAY PRODUCTS
32 CONTINUE
WRITE INFORMATION ON DECAY PRODUCTS
2
KEY2
1
35 CONTINUE
50 CONTINUE
EXIT

The program PHYHE uses a short subroutine called BETWN which simply decides whether a number is below, within, or above some upper and lower limits. The listings for BETWN and PHYHE follow.

```
      SUBROUTINE BETWN (J,K,L,M)
C     SUBROUTINE BETWN CHECKS WHETHER VALUE OF VARIABLE J LIES
C     BETWEEN VALUES OF VARIABLES K AND L.
      M=1
      IF(J-K)5,3,2
    2 IF(L-J) 4,3,3
    3 M=2
      GO TO 5
    4 M=3
    5 RETURN
      END
```

Program: PHYHE

```
      DEFINE FILE 10(33,16,U,NEXT),1(24,4,U,NREC),2(32,4,U,NREC)
      DIMENSION MO(3),IQ(3),MINM(3),MAXM(3),LP2(3,4),LP3(3,3,4),
      IPN(15),NPR(3),TM(3),JCS(3)
C     READ INITIAL EXPERIMENTAL CONDITIONS.
    1 READ (2,201)M1,IQ1,KE, MO
      WRITE (3,301)M1,IQ1,KE,MO
      DO 3 IOUT=1,3
      IF(MO(IOUT)-137) 3,2,3
    2 WRITE (3,300)
      MO(IOUT) = 0
    3 CONTINUE
      NR=24
      IP=1
      IF(137+(IQ1*M1)) 12,5,12
   12 NR = 32
      IP = 2
      IF(496+(IQ1*M1)) 99,5,99
    5 MINE = KE - (KE/10)
      MAXE = KE
      DO 6 J=1,3
      MINM(J) = MO(J) - (MO(J)/20)
    6 MAXM(J) = MO(J) + (MO(J)/20)
      INDX =1
C     FIND PARTICLES WHICH SATISFY SCAN CONDITIONS.
      DO 8 NRC=1,33
      READ(10'NRC)M,IC,T,LP2
      DO 8 J=1,3
      CALL BETWN (M,MINM(J),MAXM(J),KEY)
      GO TO (8,7,8),KEY
    7 IPN(INDX)=NRC
      INDX=INDX+1
    8 CONTINUE
```

```
      MAXNO=INDX-1
C     FIND REACTIONS PRODUCING THESE PARTICLES.
      KEY1=1
      DO 35 IB=1,NR
      KEY2=1
      READ (IP'IB)KET,NPR
      CALL BETWN(KET,MINE,MAXE,KEY)
      GO TO (35,21,36),KEY
   21 DO 28 IL=1,MAXNO
      DO 28 IM=1,3
      IF(IPN(IL)-IABS(NPR(IM)))28,25,28
   25 IF(NPR(IM)) 27,28,27
C     BRANCH AT 27 WHEN PARTICLE NO. IM OF REACTION IB IS IDENTIFIED AS
C     ONE OF THOSE SOUGHT BY THE SCAN SPECIFICATIONS.
   27 GO TO (41,42),KEY1
   28 CONTINUE
C     WHEN NO PRIMARY PARTICLES SATISFY SCAN CONDITIONS CHECK TO
C     FIND WHETHER DECAY PARTICLES ARE DETECTED BY SCAN.
      KEY2 = 3
      DO 32 IM=1,3
      KEY3 = 1
      NO = IABS(NPR(IM))
      JCS(IM) = NPR(IM)/NO
      IF(NO) 32,32,129
  129 READ(10'10) M,IC,T,((LP3(IM,MM,NN),MM=1,3),NN=1,4)
      DO 32 IL=1,MAXNO
      DO 32 NN=1,4
      DO 132 M2=1,3
      IF(IABS(LP3(IM,M2,NN))-IPN(IL)) 132,30,132
C     BRANCH AT 30 WHEN DECAY PRODUCT LP3 SATISFIES SCAN CONDITIONS.
   30 GO TO (41,130,41),KEY2
  130 KEY2 = 2
      GO TO (230,330), KEY3
  230 WRITE(3,305) IM
      KEY3 = 2
C     WRITE INFORMATION ON DECAY PARTICLES.
  330 KJ=IM
      I2=NN
      GO TO 146
  132 CONTINUE
   32 CONTINUE
   35 CONTINUE
   36 GO TO (97,98),KEY1
C     WRITE INFORMATION ON PRIMARY PARTICLES FOR SATISFACTORY REACTION.
   41 WRITE (3,302)
      KEY1=2
   42 KJM = 0
      DO 44 KJ=1,3
      NO= IABS(NPR(KJ))
      IF(NO) 44,44,242
  242 READ(10'NO)M,IC,T,((LP3(KJ,I1,I2),I1=1,3),I2=1,4)
      TM(KJ) = T
      JCS(KJ) = NPR(KJ)/NO
      JQ = IC*JCS(KJ)
      KJM = KJM + 1
      WRITE (3,304) KJ,JQ,M
   44 CONTINUE
```

```
C     WRITE OUT DATA ON DECAYS OF EACH PRIMARY PARTICLE.
      GO TO (143,130,130),KEY2
  143 DO 50 KJ=1,KJM
      WRITE (3,305) KJ
      DO 50 I2=1,4
  146 I1M=0
      DO 49 I1=1,3
      ICS = JCS(KJ)
      MO(I1)=0
      IQ(I1)=0
      NUM=LP3(KJ,I1,I2)
      IF(NUM) 46,49,47
   46 ICS=NUM/IABS(NUM)  * ICS
      NUM=IABS(NUM)
   47 READ (10'NUM)M,IC,T,LP2
      IF (M) 49,49,48
   48 MO(I1)=M
      IQ(I1)=IC*ICS
      I1M = I1
   49 CONTINUE
      IF(I1M)232,232,149
  149 WRITE (3,306)TM(KJ),(IQ(I1),MO(I1),I1=1,I1M)
  232 GO TO(50,132),KEY2
   50 CONTINUE
      GO TO 35
   97 WRITE (3,303)
   98 CALL EXIT
   99 WRITE (3,307)
      CALL EXIT
  201 FORMAT(6I5)
  300 FORMAT ('0 ** PI MESONS ARE TOO NUMEROUS FOR A SCANNING REPORT'/
     1 '      THE SCANNERS ARE IGNORING YOUR REQUEST' )
  301 FORMAT('1  INITIAL EXPERIMENTAL CONDITIONS'/
     1'   INCIDENT PARTICLE MASS='I5,' MEV/C2  WITH CHARGE='I3/
     2'0  KINETIC ENERGY OF PARTICLE BEAM ='I5,' MEV'/
     3'0  SCAN IS FOR MASSES 'I5,','I5,', AND 'I5,' MEV/C2')
  302 FORMAT ('0  SUCCESS.   RESULTS FOLLOW.'/)
  303 FORMAT('0  SCAN HAS DETECTED NO PARTICLES WHICH SATISFY YOUR CONDI
     1TIONS.')
  304 FORMAT('  PARTICLE ('I1,')   CG='I3,'   MASS='I5)
  305 FORMAT('0 DECAY PRODUCTS OF PARTICLE ('I1,') FOLLOW.')
  306 FORMAT('         DECAY TIME = 'E9.2,'  CG = 'I2,'  MASS = 'I5/
     131X,'CG = 'I2,'  MASS = 'I5/31X,'CG = 'I2,'  MASS = 'I5)
  307 FORMAT ('0  SORRY.'/'   THE LABORATORY IS NOT ABLE TO PERFORM THE
     1REQUESTED EXPERIMENT.')
      END
```

COMPUTER SIMULATION OF A MASS SPECTROMETER

Thomas R. Harbron

and

Charles W. Miller

Anderson College
Anderson, Indiana

INTRODUCTION

Every college physics department desires the best and widest range of equipment possible for its students to work with. For departments with reasonably large enrollments this ideal may be at least partially met. For departments with very small enrollments, however, the purchase of many pieces of major equipment cannot be economically justified. One answer to this problem is the simulation of physics instruments on a digital computer. The purpose of this paper is to describe how one such instrument, a mass spectrometer, has been simulated on the Anderson College IBM 1620 computer, and how it is being utilized by the physics department.

The program, MSSIM, is essentially a simulation of the Ealing Small Mass Spectrometer as described in Ealing Corporation's 1969 Teaching Catalog [also see J. W. Dewdney, American Journal of Physics 28, 452 (1960)]. Ions from a source, with some thermal kinetic energy, are accelerated through a potential and deflected in a magnetic field to a detector which measures the ion current. The problem is to find the charge-to-mass ratio of the ions observed, and from this information the student tries to determine the isotope which is observed.

The instructor inputs on punched cards the "sample" to be analyzed by the model. This sample consists of:

1. The mass of each isotope present.
2. The degree of ionization of each isotope.
3. The relative abundance of each possible ion.

He also inputs the value of the average initial kinetic energy of the ions, and initializes the random number generator to provide the "noise" current.

The remaining variables are then input by the student on the console typewriter. These are:

1. Strength of the magnetic field.
2. Range of voltage to be swept.
3. Increment to be used in sweeping the voltage.
4. Width of the detector slit.

The computer calculates an ion current for each value of voltage as it sweeps over the given range. The results are output on punched cards showing the particular voltage and the ion current associated with it. These cards may then be used as input to a program which plots a graph of current versus voltage utilizing a CALCOMP plotter.

Anderson College had no course in session during the semester in which this program was written for which it would have been appropriate to utilize this simulator, but this model will be used in the Modern Physics laboratory to be offered Semester I, 1969-70. Consequently, the Student Manual is not in its final form. In order to test the model under classroom conditions, however, an upper division student was engaged to perform two experiments with it.

The purpose of the first experiment was to study dispersion and the results of using various detector slit widths. A plot of the expected graph of mass versus accelerating voltage was made and compared to the experimental analysis of a sample containing 93%K^{39} and 7%K^{41}, all isotopes singly ionized. By varying the slit width, different size detector current peaks were obtained (see Teacher's Guide).

In the second experiment the student was to identify an unknown element. An initial sweep was made to determine the location of the different ion peaks using a narrow slit width, and then localized sweeps were made of each peak using a wider slit width to determine the relative abundance of the isotopes present. The wider slit width was used in order to obtain flat-topped peaks which were more accurate to measure than the more pointed peaks. In this case the unknown was boron, and the sample consisted of 18% singly-ionized B^{10}, 1% doubly-ionized B^{10}, 77% singly-ionized B^{11}, and 4% doubly-ionized B^{11}.

The results of these experiments proved to be quite satisfactory for both student and instructor. The first experiment led the student to a clearer picture of dispersion; the second gave him a better understanding of the use of the mass spectrometer as an analytical tool, as well as valuable experience in handling mass spectrometer data. Also, the student seemed to enjoy the experiment, especially the challenge of determining the unknown, which he did successfully. These conclusions are based both on the written report turned in by the student and conversations with him.

The instructor was also satisfied with the outcome of the experiment and felt that the goal of giving the student some feel for using a mass spectrometer without a large monetary investment was achieved. This experiment showed that it was feasible to use this simulator within the normal three-hour period assigned to the Advanced Physics Laboratory course at Anderson College. In terms of actual computer time, these runs averaged four minutes per graph to calculate and 13 minutes per graph to plot. It is expected that other experiments will be planned in the future to allow for even greater utilization of this simulator.

STUDENT MANUAL

Theory

The mass spectrometer model to be simulated in our experiment is based upon the apparatus shown in Figure 1. The ions are produced in the ion source, and ejected with thermal kinetic energies ranging from 0 to $2\Delta V$ electron volts. They are then accelerated through a potential of V volts to the source slits, which are assumed to be of infinitesimal width, where they are collimated. They now have kinetic energy (neglecting thermal energies)

$$\tfrac{1}{2}mv^2 = qV \qquad (1)$$

where m,q,v are the mass, charge and speed of the ion. The ions are then deflected in a magnetic field, B. All ions that experience the same deflection and are brought together at an image point have the same momentum, given by:

$$mv = qBr \qquad (2)$$

where r = radius of curvature of the path followed by the ion. These two conditions are simultaneously satisfied by ions with the same charge-to-mass ratio,

$$\frac{q}{m} = \frac{2V}{B^2r^2} \qquad (3)$$

thus the radius is given by:

$$r = \sqrt{2mV/B^2q} \qquad (4)$$

A radius of $r = R_0$ will bring the ions into the center of the exit slit, which has a width S. Ions passing through the exit slit will produce a measured current I_d at the detector.

The focusing of such an instrument is not perfect as there is an angular spread in the beam of any particular type of ion as it leaves the magnetic field, due to thermal kinetic energy. As a result, the actual ion current reaching the detector is found by integrating over the width of the image slit, that is:

$$I_d = \int_{\substack{\text{Slit}\\\text{Width}}} J(r)\,dr + I_n \qquad (5)$$

where $J(r)$ is the linear density dI_s/dr of true ion current over the slit width as a function of radius of curvature, and I_n is the random "noise" current of the system.

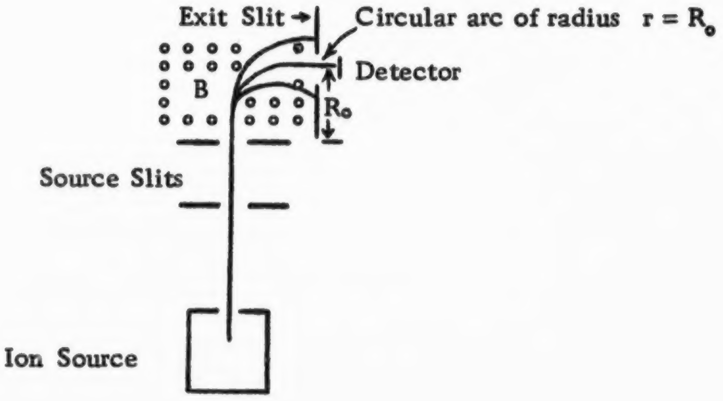


Figure 1

In this model, the component of source current due to each kind of ion present is determined by the following function:

$$I_i = \frac{A_i Q_i}{\sum\limits_{i=1}^{n} A_i Q_i} I_0 \tag{6}$$

where A_i = relative amount of given ion in sample, Q_i = relative charge of that ion, I_0 = total amount of beam current leaving source slit, 5×10^{-9} amperes.

The ions will have a few eV of kinetic energy before they are accelerated. While this energy is best described by a one dimensional Maxwellian distribution, a triangular function is assumed in the model. It is further assumed that a dispersion of energy is equivalent to a dispersion of accelerating potential. By differentiating r in Eq. (4) with respect to V, it is possible to find the difference in r, Δr, caused by a change, ΔV, in V:

$$\Delta r = \left[\frac{2m}{B^2 q}\right]^{1/2} \left[\frac{\Delta V}{2V^{1/2}}\right] \tag{7}$$

The detector will measure the current caused by all ions falling between $R_0 + S/2$ and $R_0 - S/2$. This component of detector current is determined by the following:

$$I_s = \int_{R_0-S/2}^{R_0+S/2} J(r)\,dr \tag{8}$$

For $S \ll R_0$ then $R_0 - S/2 \le r \le R_0 + S/2$ gives radii of curvature for which ions will enter the slit. The integral in Eq. (8) is then evaluated for one of nine different cases, depending on which domains of definition of the equations of $J(r)$ contain the limits of the integral.

The total ion current passing through the slit (I_s) is found by summing the contribution made by all ions:

$$I_s = \sum_{i=1}^{n} I_{si} \tag{9}$$

The total detector current is then:

$$I_d = I_s + I_n \tag{10}$$

where I_n is a random noise component.

Simulation

Parameters for the model fall in three categories as described below:

1. Parameters set by the model:
    a. Exit slit radius (R) = 4.00×10^{-2} meters.
    b. Total beam current at source slit, $I_0 = 5.0\times10^{-9}$ Amps.
    c. The variation in kinetic energies of ions leaving the ion source (assumed to have a triangular distribution).

2. Parameters set by the instructor:
    a. The mass m_i (in AMU), charge Q_i (in electron units), and relative quantity A_i of each kind of ion produced by the ion source.
    b. The maximum variation in ion energy (ΔV) in electron volts (eV).
    c. The peak noise current at the detector, I_{nmax}

3. Parameters set by the student:
    a. The value of B may be set at 1500 or 3000 gauss.
    b. The upper and lower limits of acceleration voltage (V).
    c. The increments in V for each measurement.
    d. The exit slit width (S) in millimeters.

A message from the computer will advise you to enter these parameters from the typewriter console in the following format:

Columns 1-14, magnetic flux density, B
15-28, exit slit width, S
29-42, lower limit of accelerating voltage, VL
43-56, upper limit of accelerating voltage, VU
57-70, increment of accelerating voltage, VINC

The output of the model consists of pairs of measurements of acceleration potential V and corresponding detector current, I_d. These values are printed out together with control cards for plotting graphs of I_d versus V on the CALCOMP plotter. These graphs will appear as isolated peaks. For very small slit widths, S, the peaks will be very sharp; for wide slits the peaks will be flat-topped "mesas" and the irregularities in the plateau of the mesa will be a measure of the random noise current I_n which is to be found in every electronic device. From this you may estimate the random error in detector current measurements due to noise.

Due to the thermal motions of the ions leaving the source there will be a dispersion of values of I_d about the peak value of V so that the peak whether sharp or flat will not be abrupt, but will have sloping sides resembling the so-called "Gaussian distribution." To understand this remember that V only determines the curvature of the ion path, and not the actual number of ions present.

Begin your experiment with an initial sweep using a narrow slit width, e.g., S = 0.2 mm, to determine the charge-to-mass ratios of the different ions present. Then repeat the run with localized sweeps and a wider slit width to determine the relative masses and abundances of isotopes of a given element whose presence will be indicated in the initial run by the appearance of closely-spaced sharp peaks. Hint: what do you know concerning the Q_i of isotopes of the same element? You can also determine the component of source current due to each type of isotope present from Eq. (6). Use of a wider slit will enable you to determine average peak values of $(I_d)_{avg} = I_s$ more accurately from the mesas.

What ions do you think are present in your unknown source, and in what abundances? (Your periodic table of the elements should be of some assistance in deciding what you actually have from the charge-to-mass ratios.)

The ions may have a few eV of kinetic energy before they are accelerated. While this energy is best described by a one dimensional Maxwellian distribution, a triangular distribution function f(e) for current is assumed here to simplify calculations. This function is shown in Figure 2 and described as follows:

$$f_i(e) = 0 \; ; \quad e < 0 \;, \quad e > 2\Delta e \tag{11}$$

$$f_i(e) = \frac{I_i e}{(\Delta e)^2} \; ; \quad 0 \leq e \leq \Delta e$$

$$f_i(e) = \frac{I_i}{\Delta e}\left[2 - \frac{e}{\Delta e}\right] \; ; \quad \Delta e \leq e \leq 2\Delta e$$

Note that the total area under f_i must equal the total current due to the i-th ion:

$$I_i = \int_{-\infty}^{\infty} f_i(e)de \tag{12}$$

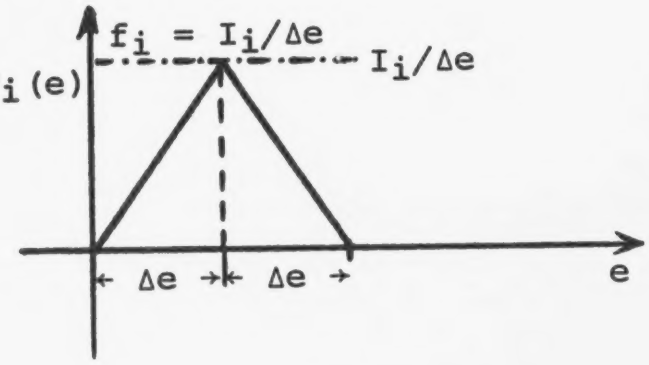


Figure 2. Triangular thermal distribution.

A further simplifying assumption is made concerning the energy distribution: that a dispersion of energy (Δe) is equivalent to a dispersion of accelerating potential (ΔV), hence, from Eq. (7) the difference in r, ΔR, caused by a change in kinetic energy $\Delta e = \Delta V$ is

$$\Delta R = \left[\frac{2m_i}{B^2 q_i}\right]^{1/2} \left[\frac{\Delta V}{2V^{1/2}}\right] \tag{13}$$

The detector will measure the current caused by all ions falling between $R_0+S/2$ and $R_0-S/2$. This component of detector current is determined by the following:

$$I_S = \int_{R_0-S/2}^{R_0+S/2} J(r)dr \tag{14}$$

where $J(r)$ is the linear ion current density as a function of r for a given value of B and V. $J(r)$ must, like $f(e)$, be a series of triangular functions $J_i(r)$, each centered about its appropriate mean radius RM as shown in Figure 3.

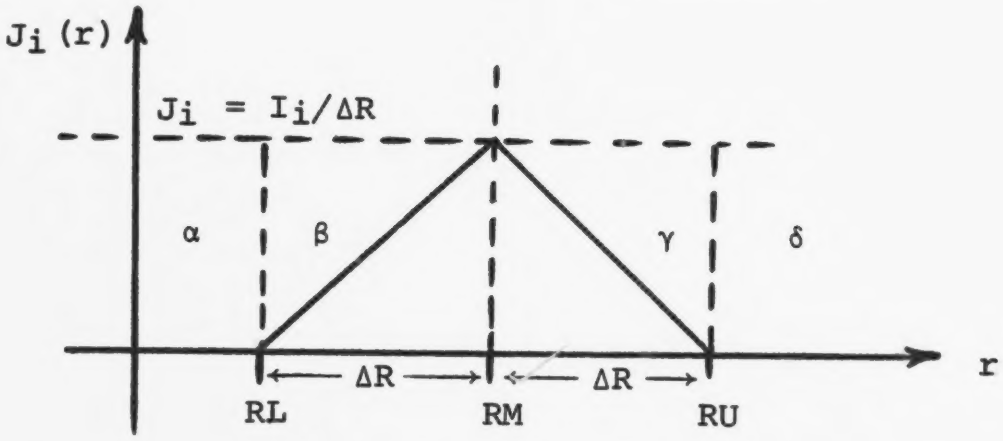


Figure 3. Triangular distribution for $J_i(r)$.

It may be found, for one ion, by transforming the preceding equations to:

$$\begin{aligned} J_i(r) &= 0 \;; \quad r < RL \;, \quad r > RU \\ J_i(r) &= \frac{I_i}{(\Delta R)^2}(r-RL) \;; \quad RL \le r \le RM \\ J_i(r) &= \frac{I_i}{(\Delta R)^2}(RU-r) \;; \quad RM \le r \le RU \end{aligned} \tag{15}$$

where

$$\begin{aligned} RL &= \left[\frac{2m_iV}{B^2q_i}\right]^{1/2} \\ RU &= \left[\frac{2m_iV}{B^2q_i}\right]^{1/2} \left[1 + \frac{\Delta V}{V}\right] \\ \Delta R &= \tfrac{1}{2}(RU-RL) \\ RM &= \tfrac{1}{2}(RU+RL) \end{aligned} \tag{16}$$

The integral given in Eq. (14) may be evaluated for nine different cases, depending on the region of Figure 3 subtended by the exit slit. That is, it depends on which parts of the distribution function are actually involved in the evaluation of the integral in Eq. (14). If we distinguish four regions, α, β, γ, δ, as shown in Figure 3, we can classify these cases mathematically according to the regions where the edges of the slit fall. Thus, the upper row of Table 1 specifies whether the rightmost edge of the slit at $r = R_0+S/2$ occurs in region α, β, γ, or δ. Similarly, the leftmost column specifies the region in which the leftmost edge of the slit at $r = R_0-S/2$ occurs, with the obvious restriction (X's in the Table) that it must occur only to the left of the other edge of the slit. This makes a total of ten separate cases in all, two of which are equivalent in that there is no overlap when the slit is either completely in region α or completely in region δ. In all cases, except when the distribution is completely contained by the slit (25), $I_{si} < I_i$. Mesas will be observed for $S > 2\Delta R$.

upper limit → lower limit ↓	$(R_0+S/2)\le RL \le (R_0+S/2) \le RM \le (R_0+S/2) \le RU \le (R_0+S/2)$			
	α	β	γ	δ
α: $(R_0-S/2)\le RL$	(17)	(20)	(23)	(25)
β: $RL\le (R_0-S/2)\le RM$	X	(18)	(21)	(24)
γ: $RM\le (R_0-S/2)\le RU$	X	X	(19)	(22)
δ: $RU\le (R_0-S/2)$	X	X	X	(17)

Table 1. Equations (in parentheses) applicable for each of nine cases of slit-distribution overlap.

The appropriate equations are as follows:

$$I_{si} = 0 \tag{17}$$

$$I_{si} = \frac{I_i S}{(\Delta R)^2}\,(R_0-RL) \tag{18}$$

$$I_{si} = \frac{I_i S}{(\Delta R)^2}\,(RU-R_0) \tag{19}$$

$$I_{si} = \frac{I_i}{2(\Delta R)^2}\,[RL-(R_0+S/2)]^2 \tag{20}$$

$$I_{si} = I_i - \frac{I_i}{2(\Delta R)^2}\{[RU-(R_0+S/2)]^2 + [(R_0-S/2)-RL]^2\} \tag{21}$$

$$I_{si} = \frac{I_i}{2(\Delta R)^2}\,[RU-(R_0-S/2)]^2 \tag{22}$$

$$I_{si} = I_i - \frac{I_i}{2(\Delta R)^2}\,[RU-(R_0+S/2)]^2 \tag{23}$$

$$I_{si} = I_i - \frac{I_i}{2(\Delta R)^2}\,[RL-(R_0-S/2)]^2 \tag{24}$$

$$I_{si} = I_i \tag{25}$$

The total ion current passing through the slit (I_s) is found as follows:

$$I_s = \sum_{i=1}^{n} I_{si} \qquad (26)$$

thus summing the contribution made by all ions.

The total detector current I_d is given by Eq. (17):

$$I_d = I_s + I_n \qquad (27)$$

where I_n is a random noise component. This component will vary between 0 and I_{nmax}, and the distribution will be uniform. The curves obtained in the trial runs are shown in Figures 4(a), (b), (c) and (d) (see pages 117 and 118).

Detailed program specifications for MSSIM are listed below and on the following pages.

Program Specifications

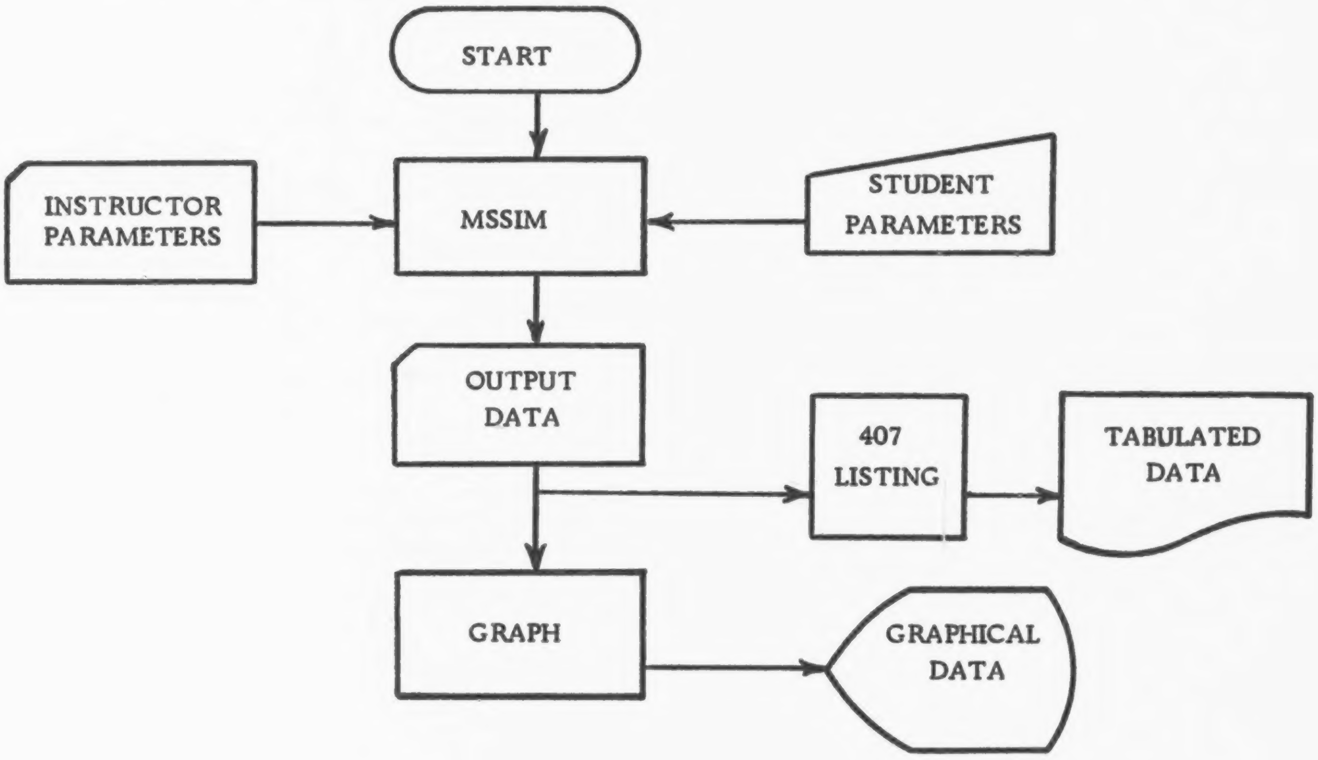
A. Name

MSSIM - Mass Spectrometer SIMulator

B. Purpose

This program simulates the operation of a mass spectrometer and generates corresponding output data.

C. System Flow Chart


START
INSTRUCTOR PARAMETERS
MSSIM
STUDENT PARAMETERS
OUTPUT DATA
407 LISTING
TABULATED DATA
GRAPH
GRAPHICAL DATA

D. Control Devices

Control cards as follows:

```
‡‡JOB           MONITOR B
‡‡XEQSMSSIM              3 (Col. 28)
```

E. Input Data

Instructor parameters:

Card 1	Columns	1-14	Initial KE of ions
		15-28	Peak detector noise
		29-34	Number of kinds of ions
		35-39	4 digit seed for random number generator
Cards 2 to N+1	Columns	1-14	Mass of ion
		18	Degree of ionization
		19-32	Relative abundance of ion in percent

Student parameters:

Columns	1-14	Flux density
	15-28	Exit slit width
	29-42	Lower limit acceleration voltage
	43-56	Upper limit acceleration voltage
	57-70	Accelerating voltage increment

See program listing and comments for complete description of input data and formats.

F. Output Data

For each run, the following cards are punched:

1. One control card for GRAPH.
2. One data card for each measurement made in the run. Each card contains the values of accelerator voltage and corresponding detector current.
3. Three data cards for GRAPH.

G. Non-Error Messages

1. Message: MASS SPECTROMETER SIMULATOR TRH.
   Cause: Identification.
2. Message: TYPE 5 PARAMETERS USING F OR E FORMATS BELOW HEADINGS B, S, VL, VU, VINC.
   Cause: Need student parameters.
   Response: Enter parameters.
3. Message: END OF RUN.
   SS1 ON TO RE-ENTER STUDENT PARA,
   SS2 ON TO RE-ENTER INST. PARA, OFF TO EXIT.

Cause: End of run.
Response: Set switch, push start.

H. Error Messages

1. Message: Parameter outside limits.
   Cause: A student parameter is outside allowed limits -
   (a) B must equal 0.1500 or 0.3000
   (b) S must be less than 10^{-2}, but positive
   (c) $500 \geq VU > VL \geq 0$
   (d) $(VU-VL) \geq VINC$
2. All standard Fortran errors are possible but should not occur. Re-check input data.

I. Switch Settings

	DISK	PARITY	I/O	O'FLOW	1	2	3	4
ON					*	*		*
OFF	X	X	X	X	*	*	X	*

*1. See non-error message 3.
*2. See non-error message 3.
*4. Used to correct typing errors.

J. File Used

None.

Bibliography

Kaplan, Irving, Nuclear Physics, Addison-Wesley Publishing Co., Inc., Reading, Massachusetts (1963).

Rusk, Rogers D., Introduction to Atomic and Nuclear Physics, Meredith Publishing Co., New York (1964).

Wehr, M. Russell and James A. Richards, Jr., Introductory Atomic Physics, Addison-Wesley Publishing Co., Inc., Reading, Massachusetts (1962).

Weidner, Richard J. and Robert L. Sells, Elementary Modern Physics, Allyn and Bacon, Inc., Boston, Massachusetts (1960).

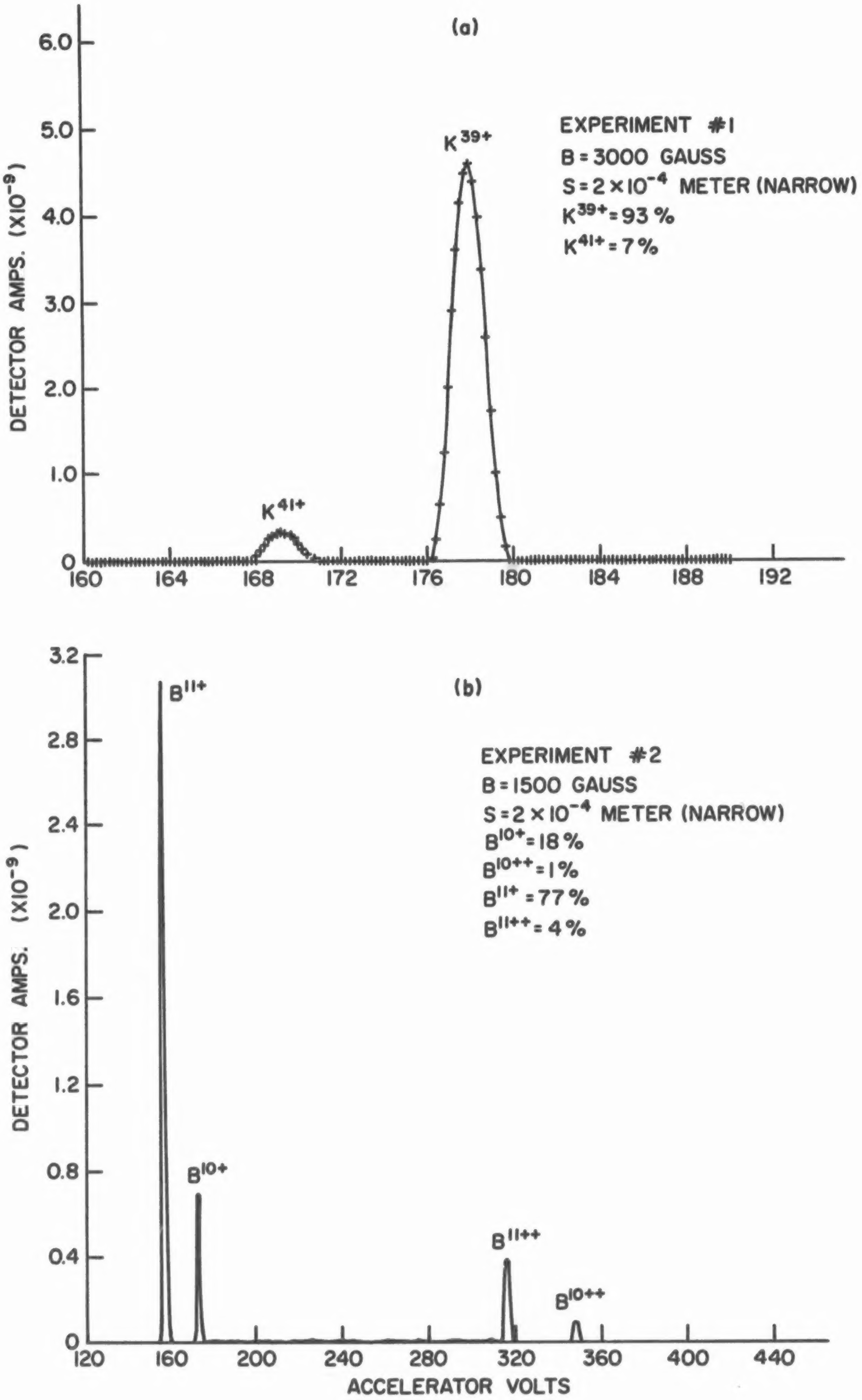
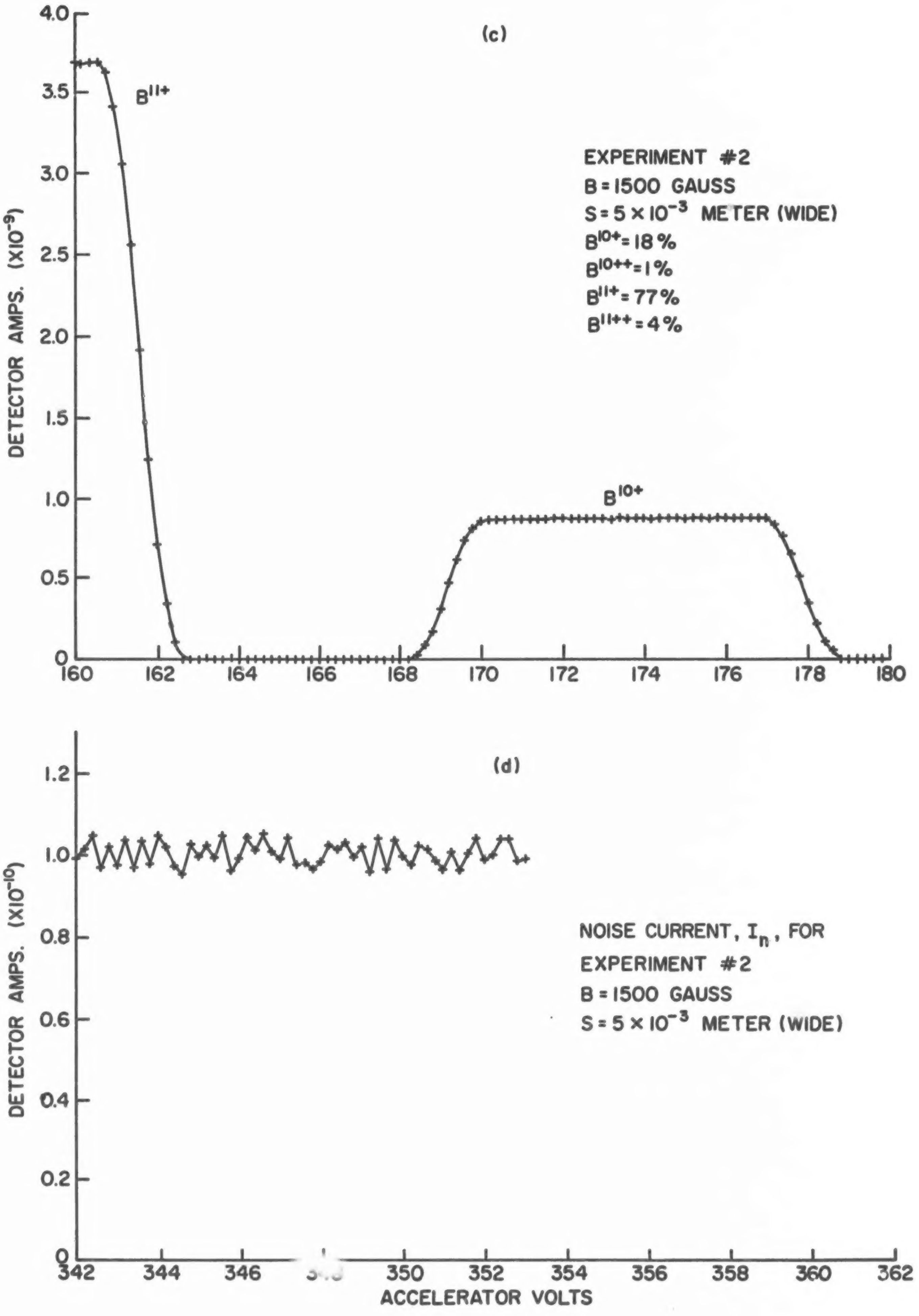

(a)
DETECTOR AMPS. (X10^{-9})
6.0
5.0
4.0
3.0
2.0
1.0
0
160
164
168
172
176
180
184
188
192
K^{41+}
K^{39+}
EXPERIMENT #1
B = 3000 GAUSS
S = 2 × 10^{-4} METER (NARROW)
K^{39+} = 93 %
K^{41+} = 7 %
(b)
DETECTOR AMPS. (X10^{-9})
3.2
2.8
2.4
2.0
1.6
1.2
0.8
0.4
0
120
160
200
240
280
320
360
400
440
ACCELERATOR VOLTS
B^{11+}
B^{10+}
B^{11++}
B^{10++}
EXPERIMENT #2
B = 1500 GAUSS
S = 2 × 10^{-4} METER (NARROW)
B^{10+} = 18 %
B^{10++} = 1 %
B^{11+} = 77 %
B^{11++} = 4 %

Fig. 4



EXPERIMENT #2
B = 1500 GAUSS
S = 5 × 10^{-3} METER (WIDE)
B^{10+} = 18 %
B^{10++} = 1 %
B^{11+} = 77 %
B^{11++} = 4 %

NOISE CURRENT, I_n, FOR
EXPERIMENT #2
B = 1500 GAUSS
S = 5 × 10^{-3} METER (WIDE)

Fig. 4

```
C  MASS SPECTROMETER SIMULATION PROGRAM
C
C  TABLE OF VARIABLES
C     RO     CENTER RADIUS OF EXIT SLIT IN METERS
C     AS     SOURCE BEAM CURRENT IN AMPS.
C     DV     AVERAGE INITIAL KE OF IONS IN ELECTRON VOLTS
C     ANMAX  PEAK DETECTOR NOISE CURRENT IN AMPS.
C     N      NUMBER OF DIFFERENT KINDS OF IONS
C     M      SEED FOR RANDOM
C     AMU    MASS OF ION IN AMU
C     Q      ION CHARGE IN ELECTRON UNITS
C     AMT    RELATIVE QUANTITY OF ION
C     B      MAGNETIC FLUX DENSITY IN WEBERS/SQ METER
C     S      EXIT SLIT WIDTH IN METERS
C     VL     LOWER ACCELERATING VOLTAGE LIMIT
C     VU     UPPER ACCELERATING VOLTAGE LIMIT
C     VINC   ACCELERATING VOLTAGE INCREMENT
C     DI     DETECTOR CURRENT IN AMPS.
C     V      ACCELERATING POTENTIAL
C     CMR(I) CHARGE-TO-MASS-RATIO FOR ION (I) IN COULOMBS/KG
C     A(I)   SOURCE CURRENT COMPONENT DUE TO ION (I)
C     DR     DELTA R
C     CMU    UPPER LIMIT OF CHARGE-MASS RATIO OF IONS HITTING SLIT
C     CML    LOWER LIMIT OF CHARGE-MASS RATIO OF IONS HITTING SLIT
C
      DIMENSION A(20),CMR(20),RX(2),K(2)
      EQUIVALENCE (RX(1),ROM),(RX(2),ROP)
C FORMAT STMTS FOLLOW -
C   ID MSG
  900 FORMAT(33HMASS SPECTROMETER SIMULATOR   TRH)
C   INSTRUCTOR PARAMETER CARD
  901 FORMAT(2E14.7,2I5)
C   ION DESCRIPTOR CARD
  902 FORMAT(E14.7,3X,F1.0,E14.7)
C   STUDENT INPUT MSG
  903 FORMAT(53HTYPE 5 PARAMETERS USING F OR E FORMATS BELOW HEADINGS/70
     1H(-----B------)(-----S------)(-----VL-----)(-----VU-----)(----VINC
     2----))
C   STUDENT PARAMETERS
  904 FORMAT(5E14.7)
C   ERROR MSG
  905 FORMAT(24HPARAMETER OUTSIDE LIMITS)
C   OUTPUT CARD
  906 FORMAT(2E14.7)
C   TERMINAL MSG
  907 FORMAT(10HEND OF RUN/74HSS1 ON TO RE-ENTER STUDENT PARA, SS2 ON TO
     1 RE-ENTER INST. PARA, OFF TO EXIT)
C   GRAPH CONTROL CARD
  908 FORMAT(18H 1 1  10.   8. 1 4)
C   GRAPH EOD CARDS
  909 FORAMT(6H-9999./18H ACCELERATOR VOLTS/17H   DETECTOR AMPS.)
C
C INITIALIZING SECTION
C    SET MODEL PARAMETERS
```

```
C
      TYPE 900
      RO= 4.0E-02
      BO= 1.5E-01
      AS= 5.0E-09
C
C READ INSTRUCTOR PARAMETERS
C
    1 READ 901,DV,ANMAX,N,M
      AT=0.
      DO 2 I=1,N
      READ 902,AMU,Q,AMT
      CMR(I)= ((1.609E-19)*Q)/(AMU*1.65979E-27)
      A(I)=Q*AMT
    2 AT=AT+A(I)
      DO 3 I=1,N
    3 A(I)=(A(I)*AS)/AT
C
C READ STUDENT PARAMETERS
C
    4 TYPE 903
      ACCEPT 904,B,S,VL,VU,VINC
      IF(B-BO)5,7,5
    5 IF(B-BO-BO)13,7,13
    7 IF(S)13,13,8
    8 IF(S-1.E-02)9,13,13
    9 IF(VU-VL)13,13,10
   10 IF(VL)13,11,11
   11 IF(VU-500.)12,12,13
   12 IF((VU-VL)-VINC)13,14,14
   13 TYPE 905
      GO TO 4
   14   ROP=RO+S/2.
        ROM=RO-S/2.
      V=VL
C
C   PUNCH GRAPH CONTROL CARD
      PUNCH 908
C
C  MAIN PGM LOOP TO CALCULATE DI FOR EACH VALUE OF V
C
   15 DI=0
      CML=(2.*V)/(ROP*B)**2
      CMU=(2.*(V+2.*DV))/(ROM*B)**2
C  CHECK EACH ION TO SEE IF IT CONTRIBUTES TO DI
      DO 50 I=1,N
      IF(CMR(I)-CMU)17,17,50
   17 IF(CMR(I)-CML)50,18,18
C  PROCESS ION IF IT CONTRIBUTES TO DI
   18 RL=SQRT((2.*V)/(B*B*CMR(I)))
      RU=RL*(1.+DV/V)
      DR=(RU-RL)/2.
      RM=(RU+RL)/2.
      F=A(I)/(2.*DR*DR)
```

```
C  DETERMINE WHICH EQUATION TO USE
      DO 30 J=1,2
      K(J)=0
      IF(RX(J)-RL)30,30,21
   21 IF(RX(J)-RM)22,22,23
   22 K(J)=1
      GO TO 30
   23 IF(RX(J)-RU)24,24,25
   24 K(J)=2
      GO TO 30
   25 K(J)=3
   30 CONTINUE
      J=1+K(1)+4*K(2)
      GO TO (50,50,50,50,110,108,50,50,113,111,109,50,115,114,112,50),J
C FOLLOWING EQUATIONS CALCULATE CONTRIBUTION OF ION (I) TO DETECTOR CURRENT
  108 C= F*2.*(RO-RL)*S
      GO TO 31
  109 C= F*2.*(RU-RO)*S
      GO TO 31
  110 C=F*(RL- ROP )**2
      GO TO 31
  111 C=A(I)-F*((RU-ROP)**2+(ROM-RL)**2)
      GO TO 31
  112 C= F*(RU- ROM )**2
      GO TO 31
  113 C= A(I)-F*(RU- ROP )**2
      GO TO 31
  114 C= A(I)-F*(RL- ROM )**2
      GO TO 31
  115 C=A(I)
   31 DI=DI+C
   50 CONTINUE
      CALL RANDOM(R,M)
      DI=DI+R*ANMAX
      PUNCH 906,V,DI
      V=V+VINC
      IF(V-VU)15,15,60
C
C   END OF RUN ROUTINE
C
   60 TYPE 907
C   PUNCH EOD GRAPH CARDS
      PUNCH 909
      PAUSE
      IF(SENSE SWITCH 1)4,61
   61 IF(SENSE SWITCH 2)1,62
   62 CALL EXIT
      END
```

RANDOMNESS AND RADIOACTIVE DECAY

Kenneth F. Kinsey

and

John Kenyon

State University of New York at Geneseo
Geneseo, New York

INTRODUCTION

The material presented here has been taught to an audience of biology, earth science, chemistry and mathematics majors in an introductory, noncalculus, service course in general physics. It was also presented, in preliminary form, during the summer session of 1968. At present, we are engaged in an extensive revision in approach and content, and we have not yet chosen an appropriate accompanying text.*

Roughly, the course will stress a discussion of the state of a system in general terms; conservation laws in terms of those parameters of the description of a system which are constant, and the interaction laws, in terms of those properties of a system which change. Emphasis is placed on constructing models which simulate the behavior of the actual physical system. The computer was introduced into this course for two reasons: a) many of these people will not otherwise be exposed to computers; b) we can use the computer as an aid in constructing and studying models. The lab in this course is a three-hour lab, of which one hour is a problem session.

We have not made any attempt to explain the programs themselves to the students, since one essential aspect of the technique is that the computer output be well-formatted and self-explanatory. It is also necessary, for batch processing, to trap out as many invalid or erroneous input parameters as possible--with 150 students any conceivable combination of invalid data is likely to occur. These factors add considerable complications to the essential program, and would make it quite difficult for students to follow.

We found no opposition to the "black-box" technique--a little too much faith, in fact--and plan, in the future, to introduce a lab in which we will look more closely at programming and the "Garbage In--Garbage Out" effect.

The problem illustrated in this paper originated in an experiment to measure the decay of radioactive silver produced in a neutron howitzer. The logistics of retrieving and distributing the computer output makes a division of the material into three lab periods a more workable arrangement, as explained in the Teacher's Guide.

A third simulation program is included in the package, although we did not incorporate it into the summer session course. This program simulates the buildup to saturation of a source where the production is at a constant rate. We have discussed the advisability of including a simulation of the two distinct half-lives that are present in the actual source. However, the shorter one is rather difficult to measure experimentally, except as an excess of counts at early times.

---

*Basic Physics, K. W. Ford, Blaisdel (1968) is used as a reference text.

Based on experience with the summer school class, the experimental writeups have been extensively rewritten in the interests of clarity and fuller explanation. We are greatly pleased with the results and have incorporated them into the new course at an early stage--preceded only by the PSSC "analysis of an experiment" lab. In fact, these simulation programs could be used in a more advanced course with appropriate modifications of the lab writeups. Advanced students might investigate such things as the effect of the "quantized" throws versus the continuous decay, and could use the neutron capture cross sections to develop a Monte Carlo calculation of the activity buildup.

A word about philosophy for the General Physics course. A computer is a Black Box to these students, and will probably always be so for the majority. The aim is to convert a mysterious, omnipotent Black Box into a useful Black Box. We have kept the calculations conceptually simple and, in each case, have required the student to perform the same operations on a small sample as will be done on larger samples by the computer. The programs themselves are complicated, for reasons explained above; but we believe that in order to get clear and comprehensible output, it is better to accept a complex program which students will not understand, than to write simple but inadequate programs (which most students still will not understand). At some point in the course--either in the lecture or laboratory--we will try to remove some of the mystery about programming.

The following section, entitled "Student Manual," presents material which was distributed to students participating in the two experiments. It is here reproduced in a slightly condensed version.

We would like to acknowledge the cooperation of Dr. Rex Adelberger, who has been teaching the General Physics course this summer.

Randomness and Radioactive Decay I

As we have frequently done, we will be trying to invent a model of a physical system that satisfactorily displays some of the properties of the real system. The process we will study is "radioactive decay".

1. Set up the Geiger counter (see instruction sheet) and pick the measuring time and distance from the weak source so that you get about two or three counts in an interval. Take a series of 100 "runs" and record the number of counts obtained in each. Plot a histogram of the number of runs with 0 counts, 1 count, 2 counts, etc. Calculate the average number of counts/run.

2. Examine the first half of your series of runs and see if you can detect any pattern in the number of counts/run. If you see a pattern, cover the lower half of the list and try to predict successive numbers on the basis of the preceding ones.

   Do you feel that you were any more successful than you would have been if you had just guessed?

3. Compare the average number of counts/run from the last half of your series to that from the first half. To what extent is the behavior of this system patternless?

The model we have of radioactive decay is roughly as follows:

The "source" consists of a large number of discrete nuclei; each of these nuclei is in a particular nuclear state. There is another nuclear state to which the nucleus can make a transition, and still satisfy all of the conservation rules. For most radioactive transitions, these two states correspond to nuclei of different chemical elements. Conservation of Mass-Energy decrees that the second state have a lower total Mass-Energy than the first; the excess Mass-Energy is necessary to trip the detector. Actually, we should take this fact into account, since it implies that a nucleus can undergo the transition only once. We will return to this point in a later experiment.

One of the most fundamental features of our model of the nuclear particles is that all examples of a particular nuclear state are identical and indistinguishable. This follows from our definition of "state", since two systems in the same state are, by definition, indistinguishable. Now, if two systems are truly identical, at first thought it would seem that the past and future histories of these systems must also be identical. That is, if we create a number of identical nuclear states (or any other type of state) and if one of them undergoes a spontaneous transition at some time, then they should all do it at the same time (or else they are not identical). However, the source that you have measured was "manufactured" in a nuclear reactor during a relatively short period of time. During this time, all of the nuclear systems were forced into their excited states; yet transitions, as you have observed today, continue to occur at different times after production.

This raises the possibility that the nuclei are not all identical, but that each contains a "clock," set to go off at different times. If this were the case, it might be possible to physically separate those with short decay times from those with longer times. This has not proved possible, nor is there any evidence that the transition process is different in any respect for early or late decays.

There is, however, one possibility which supports the idea of identical nuclear states. Let us imagine that the transition time of each nucleus is determined, not by a preset clock, but purely by chance, and that the probability of each nucleus undergoing transition in a given period of time is identical to that of each other nucleus.

An analogy immediately comes to mind--a batch of identical pennies. If we dump them on the floor, they each have a 50-50 chance of landing "heads". The probability, we would say, is 1/2. A batch of identical dice would each have a 1/6 probability of showing, say a "5" on the upper face. Speaking figuratively, of course, let us give each nucleus a die which it will toss at the start of each time interval to decide whether or not to make a transition. The details of how the nucleus "decides" are not interesting, if in fact that is even a valid question to ask. The point is that it "decides" purely on the basis of chance, and that the chances are the same for each nucleus in the source.

For the moment, let us talk only about the probabilities, in terms of throwing dice. We will imagine, then, each nucleus throwing a die at the start of each counting interval. Each die has one side on it which we call "heads"; the other sides are all "tails". Let us also assume that the average number of decays you observed was three. Suppose there were six nuclei throwing six-sided dice; then we would expect one head, on the average, since the probable outcome of $1/6 \times 6 = 1$. We would get three heads, on the average, from 18 nuclei throwing six-sided dice ($1/6 \times 18 = 3$); or 15 nuclei throwing five-sided dice ($1/5 \times 15 = 3$); or 3×10^{23} nuclei throwing 1×10^{23}-sided dice ($1/10^{23} \times 3 \times 10^{23} = 3$). We see that if there are enough nuclei, the chances for any given nucleus can be pretty small and still give a reasonable number of transitions.

4. Let us test the hypothesis by doing a dice-throwing experiment. Since our dice have six sides, select a number of dice which will give the same average num-

ber of "heads" as the counts you obtained with the counter. Throw your batch of dice a number of times and make a histogram of the number of times 0 heads, 1 head, 2 heads, etc. show up, just as you did with the radioactive source. Do these two sets of numbers behave the same way? Can you predict any particular number of heads? Can you predict the average?

5. Repeat the experiment with pennies, again chosen to give the same average (any problems?). Compare the two histograms.

At this point, we must recognize a deficiency in our model. We are using, say 18 six-sided dice to get an average of three, but our source is using more like (3×10^{23}) 10^{23}-sided dice to get the same average. The problem is, we will never get 20 heads (with only 18 dice), but it is certainly not out of the question if you have 3×10^{23} dice. To compare results we need to use more dice than we can conveniently handle.

To help with this problem the 1130 computer has been programmed to simulate the throwing of a large number of dice, each with N number of sides. It will also plot out a histogram showing the actual number of times heads comes up as well as the most likely number of times, based on a mathematical analysis of the probabilities.

6. Pick a number of dice (around 100) and a number of sides which will give you the average value you measured. For example, if you wanted an average of 2.5, 40 sides would be correct: $100 \times 1/40 = 2.5$. Ask the computer to throw this set of dice the same number of times as you made counting runs. Plot your data, and the computer's data on the same graph. From a comparison of the graphs, would you say that the dice-throwing model gives an accurate reproduction of the actual physical situation?

Comments on Random Numbers

Although a computer cannot actually roll dice, it can be programmed to produce patternless, or random, sequences of numbers, one through six, for example, that simulate the random way in which the numbers would appear on successive throws of a die. Not that it will duplicate the string of numbers, of course, but the sequence produced by the computer and by a real die have the following common characteristics: a) the next number in the sequence cannot be guessed from the preceding one more often than 1/6 of the time (on the average); b) in a long sequence each of the six different numbers will appear about 1/6 of the time. The major difference between the two random processes is that if one starts the computer over again, it will give the same sequence as before. However, the number N in your data will enable you to start the computer off at a different place each time, thereby avoiding repetition of the previous sequence.

Roughly speaking, the method of generating random numbers in the computer consists of multiplying N by another number Z and retaining only the rightmost (low-order) half of the *digits* in the product as a new number N'. N' is then multiplied by Z to generate a new number N" in the same way. Thus, if N = 13 and Z = 123, then NZ = 1599 and N' = 99; similarly, N" = 77. (Although the example we have chosen only yields odd two-digit numbers, the actual program in the computer is more sophisticated, yielding both odd and even numbers.)

Program Notes

The 1130 computer programs for the sequence of labs on radioactive decay are as follows.

DITHR This program simulates the throwing of a batch of N-sided dice any given number of times. The output from DITHR is largely self-explanatory. The graph is expanded to use all of the space available. The scale factor is printed on the sheet. If a scale factor of 3.00 were computed, three asterisks would correspond to one occurrence. You can replot the results to any scale by using the numbers at the right of the page. The numbers labeled as "theoretical" are what you would expect from the average of a very large number of throws. To operate this program, punch the following data cards (FREE STYLE):*

1. A title card (your name, date, etc.--whatever you like).

   This card will be reproduced on the printed output to let you identify your results.

2. A card, or cards, containing the following numbers in this sequence:

NDICE	The number of dice in the batch.
NSIDE	The number of sides on each die.
NTHROW	The number of times the batch of dice is to be thrown.
N	A starter for the generator of random numbers--a positive, odd integer less than 30,000. You should invent your own number N.
9999	

For a simple case of NDICE = 100, NSIDE = 25, and NTHROW = 100, the output appears in Figure 1 on the following page.

---

*See notes on punching data cards further in the text.

```
THERE ARE ORIGINALLY  100  DICE, EACH HAVING   25  SIDES

N=   13       PRINT FOR EVERY     3  THROWS


-------------------------------------------------------------------------------------------

                           JOHN KENYON

-------------------------------------------------------------------------------------------

INTERVAL  ACTUAL   ACTUAL   THEOR.   THEOR.   I INTERVAL ACTUAL   ACTUAL   THEOR.   THEOR.
          DICE     DICE     DICE     DICE     I          DICE     DICE     DICE     DICE
          LEFT     DECAY    LEFT     DECAY    I          LEFT     DECAY    LEFT     DECAY
                                              I
            100             100.11            I            13               15.92
    3                 15              11.52   I    48                 1               1.83
             85              88.47            I            12               14.09
    6                  7              10.19   I    51                 0               1.62
             78              78.27            I            12               12.46
    9                  6               9.02   I    54                 3               1.43
             72              69.25            I             9               11.03
   12                  7               7.98   I    57                 2               1.27
             65              61.27            I             7                9.76
   15                  8               7.06   I    60                 0               1.12
             57              54.20            I             7                8.63
   18                  6               6.24   I    63                 2               0.99
             51              47.96            I             5                7.63
   21                  5               5.52   I    66                 2               0.88
             46              42.43            I             3                6.75
   24                  5               4.89   I    69                 0               0.77
             41              37.54            I             3                5.98
   27                  8               4.32   I    72                 1               0.68
             33              33.21            I             2                5.29
   30                  6               3.82   I    75                 0               0.60
             27              29.38            I             2                4.68
   33                  5               3.38   I    78                 1               0.53
             22              25.99            I             1                4.14
   36                  1               2.99   I    81                 0               0.47
             21              23.00            I             1                3.66
   39                  3               2.65   I    84                 0               0.42
             18              20.35            I             1                3.24
   42                  5               2.34   I    87                 0               0.37
             13              18.00            I             1                2.86
   45                  0               2.07   I    90                 1               0.33
```

Figure 1

Notes on Punching Cards

Data to be read by a computer usually has to satisfy rigid requirements about columns, spaces, etc. However, data for student programs can be entered in a much more liberal fashion. This is true of those programs where data is indicated as being FREE STYLE.

For such programs, the order in which your numbers appear is important. Any given number must be entered with all digits punched together and no spaces, commas, or any other characters between digits. The one exception to this is that a decimal point will be honored if it is punched. You may not use column 80; you may not split a number between two cards. Use only eight digits per number.

As Punched	As Interpreted
95	95.
- .95	- 0.95
123,456.7	123 456.7 (2 numbers)
123456789.12	?? (more than 8 digits)
X = 15, Y = 3	15 3
9999	0

The last item on your cards should be the number 9999. This signifies no further data in this batch. (On occasion, the data for a single program may consist of more than one batch.) The number 9999 should always appear on the same card as your last data item, except when you are explicitly directed otherwise.

If you make an error in punching: back up and punch XXXX over the entire wrong entry--including sign and decimal point. Space on to a clean part of the card and punch it correctly.

If you need to use a big number: a form of scientific notation is allowed; the number 3.4192×10^8 would be punched 3.4192E+08 and the number -6.62×10^{-23} would be punched - 6.62E-23. Consult your instructor (or instruction sheets) before using this feature.

Radioactive Decay II

In the last laboratory we investigated a possible model of a radioactive source. This model assumed the source to be a large number of identical nuclear states, each with an identical probability of undergoing a transition in any interval of time. For each system in our sample, the occurrence of a transition was determined purely by chance; the mechanism for making this decision was not considered. We compared the behavior of the source with the behavior of a set of dice so constructed that the average number of "heads" agreed with the average number of counts from our source.

In making this comparison we explicitly neglected one factor--

that a nuclear state can only undergo this transition once. We can make this conclusion on the grounds of conservation of Mass-Energy, since the "triggering" of the detector by the transition must require some energy. Thus, one characteristic of the two states of the system must be that the second state has a lower Mass-Energy than the first. (This is always true of any transition that occurs spontaneously.)

A nuclear system which has undergone a transition is now in a different state and can no longer be counted among those systems which have a probability of undergoing a transition. It is, of course, possible that this new state will itself be unstable, having some probability--maybe very different--of changing to a third state, which in turn may change to a fourth state, and so on, thus forming a radioactive series. We will only consider the case where the second state is stable.

Our dice-throwing model needs only a minor modification to take this one-shot transition factor into account. Namely, whenever a die comes up "heads"--corresponding to a transition--we remove it from the sample. It is obvious that the number of dice in our sample must get progressively smaller, and that as the sample gets smaller, the number coming up "heads" will also get smaller.

Question: What does the number of counts on the Geiger counter correspond to--the number in the sample, or the number that come up heads?

1. Try the experiment with a batch of coins, tossing them repeatedly and removing at each toss those that come up heads. Plot a graph of the number in the sample before each throw and of the number coming up heads vs the number of throws. On the same graph, plot what would have happened if exactly half had come up heads at each toss. Did you run into any problems making this last plot? If so, what did you decide was meant by "half a head"? Did you get a straight line graph?

The reason you did not observe this sort of steady decrease in the counting rate in the last lab was that the probability of transition was low and the number of nuclei was large. While removing three from a sample of 10^{23} alters very little, taking three from a sample of six makes a big difference.

Our task now is to find a way of making a meaningful comparison between the data derived from our model and that which you will get from the actual radioactive sample. It is important to realize that we will not detect all of the transitions in our source, but only some fraction of them, determined by the size of our detector, its distance from the source, etc. Thus, all of our measurements are relative, and we cannot learn the actual number of nuclei or transitions in any interval. This should not bother us; we can imagine our dice-throwing experiment in the same way--as if the number of dice we started with was some unknown fraction of the total. Presumably, the part of the sample that we did not see would decay at the same fractional rate as the part we observed; i.e., 1/5 gone in time t, 1/2 in time t', etc. Thus, our comparison cannot be based on the absolute values of the data, but must test their relative behavior. That is, if we plot heads (or counts) versus time, the vertical scale of our graph is not important.

Another problem in the comparison is the horizontal scale of the graph--there is no reason why one throw of the coins or dice (whose probability of a transition we chose quite arbitrarily) should correspond to 10 seconds of time for the actual radioactive source. In other words, the time (horizontal) axis of our graph also has an arbitrary scale. Our comparison must be independent of the scale of our graph along either axis; the "shape", the fractional change in the height of the graph with time, is more important than "size".

Question: What happens to a straight line graph if you change the scale of the axes? Does the data look like some power of the time? (Inverse power?) This, you will recall, can be tested by plotting log (data) versus log (time) and seeing if a straight line results.

Try plotting log (data) versus time and see what happens. If one of these graphs gives a straight line, does it still do so if you change the scale of the axes?

2. Repeat your experiment with a batch of dice. Plot your results on the same graph that gave a straight line--compare the graphs.

   The 1130 computer has again been programmed to throw a set of dice, and to remove all of those coming up heads for the next throw. Decide on a number of dice in your sample and on the number of sides each is to have. Punch up the control cards as instructed. (See Program Notes, EXPSM.) Plot the data vs the number of throws. Also plot the number as calculated if 1/(no. of sides) of the dice were removed each time.

If we are to observe this experimentally, we shall have to create our own radioactive source. This is reasonable since, if the probability of transition is to be high enough so that an appreciable fraction of the nuclei undergo transitions during a laboratory period, there would be no possibility of keeping such a source from year to year (or even for the entire week). The radioactive source we will use is an isotope of silver, which we will produce from normal silver by bombardment with neutrons in a neutron "howitzer". This radioactive isotope will be too "hot" for you to count the transitions individually, but we can use the "count rate meter" to measure the rate at which the transitions are occurring.

3. Take a "hot" quarter and set it at such a distance from the detector that the rate meter needle is nearly at the top of its scale. Record the reading of the meter at 10-second intervals until the reading is too small to observe accurately.

   We now have two sets of data: that from the "dice" and that from the radioactive source. Compare your data to that which the computer generated, using our model, by plotting them on the same graph. Do you consider the model to be a good one?

   If you consider this to be a good model of the radioactive source, then approximately to what time interval does one throw of your dice correspond?

EXPSM

This program simulates the throwing of a batch of dice, removing at each throw those which come up "heads" (chosen to be side N for N-sided dice). The output from EXPSM, as shown, is largely self-explanatory. The number of surviving dice is printed down the page, and, staggered between them, is the number that have been removed (if we do not print every throw, then the total removed since the last throw printed is given). This is labeled with the number of the throw. The theoretical values are obtained by removing exactly 1/NSIDES of the dice on each throw. This accounts for the fractional values. The program will stop calculating when all of the actual dice have been removed, hence, NTHROW is unnecessary. To operate this program, punch the following data cards (FREE STYLE):

1. A title card (your name, date, etc.); this will appear on your printed sheet.

2. A care, or cards, containing the following numbers in this sequence:

NDICE	The number of dice originally present.
NSIDE	The number of sides on each die.
N	A starter for the random number sequence--a positive, odd integer less than 30,000.
9999	

OPTION: You may, if you wish, insert one more number--NCYCL--after N and before 9999. This number tells the computer how often to print your results. If NCYCL is 3, for example, the results after every three throws will be printed. This will result in a quicker execution of the program.

For a sample case of NDICE = 100, NSIDE = 25, and NCYCL = 3, the output is given in Figure 2, on the following page.

```
A SET OF  100   DICE,EACH DIE HAVING   25  SIDES WAS THROWN  100  TIMES

A SCALE FACTOR OF 4.00 IS COMPUTED

N=   13

------------------------------------------------------------------------------------------------------------------------

                                          HISTOGRAM OF FREQUENCY DISTRIBUTIONS

                                          JOHN KENYON AND GARY CHARLES AS PARTNERS

------------------------------------------------------------------------------------------------------------------------
                                                                                                FREQUENCY OF OCCURRENCE
NUMBER OF TIMES A SIDE
APPEARS  IN  ONE  THROW                                                                          ACTUAL    THEORETICAL

   0  .*****O******                                                                                3          1.687
   1. .***************************O************                                                   10          7.034
   2  .******************************************************** O                                 14         14.508
   3  .************************************************************                 O               15         19.747
   4  .******************************************************************************O*****        21         19.953
   5  .**************************************************************O*                            16         15.962
   6  .**************************************** O                                                  10         10.530
   7  .**********************O*                                                                     6          5.891
   8  .**********O*                                                                                 3          2.853
   9  .***O                                                                                         1          1.215
  10  .O***                                                                                         1          0.460
  11  .                                                                                             0          0.157
  12  .                                                                                             0          0.048
  13  .                                                                                             0          0.013
  14  .                                                                                             0          0.003
------------------------------------------------------------------------------------------------------------------------
```

Figure 2

TEACHER'S GUIDE

Logistics

The Computer Center on the Geneseo campus of the State University of New York has an IBM 1130 computer, which operates on a completely open-shop basis. The general physics course in which this pair of laboratory experiments was used has about 140 students in six sections. In our first experimentation with these laboratories, we made an attempt to get the computer output back to the students during the lab period by having a messenger pick up punched cards, take them to the computer, process them and bring back the printed output. This procedure did not prove successful and experience indicates that one cannot count on the computer's availability during all sessions. For this reason, this experiment package, although it constitutes two experiments, was actually distributed over three laboratory periods.

During the first period, students took the data on the background radiation with the Geiger counters, ran the simulation experiment with either dice or the pennies, and punched the cards to generate the computer data. Since we were dealing with complete novices in the art of computing, we examined all data cards for validity. Four manual card punches were used in the laboratory, a number which proved to be quite adequate for the 24 students in each section. These punches are sufficiently simple in operation and the FREE STYLE program (discussed elsewhere in this monograph by Paul Smith of Coe College) was so undemanding of the students, that almost no additional instruction on card punching proved necessary, other than that which is provided in the Student Manual. At the conclusion of the laboratory, the data cards were picked up and processed in a batch. Students retrieved their output at their convenience.

Comments and Cautions

It is quite important that the counting rate and the counting time for the Geiger counters be adjusted so that the number of counts is on the order of two or three per interval. This will result in a sufficiently skewed distribution so that valid comparisons may be made between the simulation data and the actual data. The instructor should resist the temptation to reduce the number of counting runs from 100. If the number of runs is substantially smaller than this, the statistics of the small numbers will result in distributions which are rather far from typical, making the comparisons unconvincing. In practice, the recording of the Geiger counter data did not prove to be a time-consuming task. Students encountered some difficulty in setting up the Geiger counter properly, and it proved advisable to have the voltages preset by the lab instructor. Although it does introduce an element of black box technology, the point of the lab was not the theory and operation of a Geiger counter, but rather that of a mathematical model of a physical situation.

Laboratory Period 2

The first part of this period was spent in completing the dice-tossing or coin-throwing and in discussing the comparisons of the various sets of data. At this point, we conducted a rather careful discussion of what was meant by the theoretical data. In the second part of this laboratory period, the students performed the simulation parts of the exponential decay in experiment number II. At the conclusion of this period, they punched the cards for the second computer simulation. Since our computation center operates on an open-shop basis, we decided to have the students run their own programs so that they could see the equipment actually operate and watch the output emerge. To facilitate this, the program EXPSM was stored on the 1130 disk in a core image format to minimize the loading time. We also provided a batch of prepunched JOB and XEQ cards, and the Computer Center made available a brief set of operating instructions. The experiment of having 140 novices operate their computer programs during a one-week period proved to be remarkably successful, although somewhat trying for the Computer Center personnel. If anyone is tempted to repeat this, we strongly urge that a large wall chart be posted, containing operating instructions for the computer and, most importantly, what not to do. This will minimize, though not eliminate, the reloading of the disk. The cards could, of course, be run off in a batch as was done for the first experiment. However, student response to doing it themselves is so favorable that we strongly suggest it be tried if local conditions permit.

Laboratory Period 3

During this period the students took the actual data for the decay of the radioactive silver produced in the neutron howitzer. Also, some discussion of the graphing and the comparison of the data proved to be advisable during this laboratory period.

General Comments

Throughout the three periods we took considerable pains to emphasize that the point of the experiment was the comparison of the model with the experimental situation, and to prevent the technology involved from obscuring this point. A calculated risk was taken in introducing, at the very start of the course, an experiment with this degree of sophistication. The major problem encountered related to the inexperience of the students in graphing and interpreting experimental data. Eventually, an additional lab involving data-handling will precede this package. Student response was generally favorable, but the instructor must take care to see that the objectives of the experiment do not become obscured.

Random Number Generation on the 1130 Computer--Subroutine Dice

The subroutine RANDU, in the IBM supplied Scientific Subroutine Package, generates "random" numbers by repeated multiplication, retaining the low order part of the product. The integers are reasonably well distributed, if one ignores the fact that they are all odd and that the repetition cycle of the n^{th} bit is 2^{n-2}.

For simulation and Monte Carlo purposes, one would frequently like a random integer between 1 and N, where N is small. This can be accomplished by taking a random number "modulo N" (or dividing by N and adding 1 to the remainder). The routine RANDU is ill suited to this purpose because of the poor cycle of the bits that are retained by this process.

Subroutine DICE(NSIDES,IUP,N) generates such a random integer between 1 and NSIDES, returning the value in IUP. N is the starter for the random number sequence and should be an odd, positive integer. This subroutine circumvents the difficulties of RANDU by generating a new "random" number N and then reversing the 15 significant bits before dividing by NSIDES. The small integers thus produced have the best possible cycle. Protection is also provided against the logically difficult situation of a call with NSIDES ≤ 0. The name of the program is suggested by the throwing of an N-sided die. Because of the extensive bit manipulation the program is written in Assembler language. It would be possible to write it in Fortran; however, conversion to a different machine should not be attempted, since the particular routine is hardware dependent.

If a user had a need for a more extensive string of random numbers greater than 2^{13} (8192) the routine could be revised to treat the multiplier 899 as a variable. (This number is one of a family $8t \pm 3$.)

Two auxiliary subroutines IDICE and HDICE are useful in simulation routines. IDICE (NUP,NDICE,NSIDES,N) where NUP is Dimensioned NUP(NSIDES) returns the number of times each side was up in a single throw of NDICE. HDICE(NHDS,NDICE,NSIDES,N) returns the number of "heads" in a set of NDICE, where the side called "heads" is determined by a first call of DICE.

Listed on the following pages are the source and subroutine programs for the experiments.

```
// JOB
// DUP
*DELETE             DITHR
// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
*IOCS(CARD, 1132PRINTER)
C     PROGRAM DITHR
C     WRITTEN BY J KENYON.  SUPERVISED BY K KINSEY, SUNY,GENESEO
      INTEGER BLANK,STAR,DOT,X,DASH ,O
      DIMENSION ICNT(500),T(4),ILINE(120),LINEA(96),LINET(96), IHEAD(40)
      DATA BLANK,STAR,DOT,DASH,O/' ','*','.','-','O'/
      COEF(NDICE,IK,P)=2.71828**(FCTLN(NDICE)-FCTLN(NDICE-IK)-FCTLN(IK))
     X*(P**(IK))*((1-P)**(NDICE-IK))
3     READ(2,5) IHEAD
2100  CALL FREE(T,4)
      PVAL=0.
C     NDICE IS THE NUMBER OF DICE BEING THROWN
      NDICE=T(1)
C     NSIDE IS THE NUMBER OF FACES ON THE DIE
      NSIDE=T(2)
C     NTHRW IS THE NUMBER OF TIMES NDICE ARE THROWN
      NTHRW=T(3)
C     N IS A STARTER IN THE RANDOM GENERATION OF NUMBERS
      N=T(4)
C     N MUST BE AN ODD INTEGER
      IN=N
C     CHECK THE VALIDITY OF THE DATA
C     IF NSIDE=0, READ IN NEW SET OF DATA
C     IF NSIDE=1, SET VALUES AND CONTINUE
C     IF NSIDE IS GREATER THAN 1, CONTINUE AND CALCULATE VALUES
      IF (NSIDE-1) 2200,2300,2400
2200  GO TO 3
2300  JK=NSIDE
      II=NSIDE
      SCAL=96./NSIDE
      IVAL=1
      VAL=1.0
      IX=96
      DO 2307 I=1,96
2307   LINET(I)=BLANK
      DO 2301 I=1,IX
2301  LINEA(I)=STAR
      LINET(IX)=O
      ISW=1
      GO TO 739
2302  WRITE(3,2202) JK,DOT,LINEA,IVAL, VAL
      WRITE(3,2205) LINET
      DO 2303 I=1,96
      LINET(I)=BLANK
2303  LINEA(I)=BLANK
      GO TO 803
2400  ISW=2
```

```
      KKK=0
      P=1.0/NSIDE
C     II IS THE SIDE UP WHICH WE ARE COUNTING ON EACH THROW
      II=NSIDE
      DO 1000 I=1,NDICE
1000  ICNT(I)=0
C     ICNT IS THE NUMBER OF TIMES II COMES UP ON A PARTICULAR THROW
      DO 40 K=1,NTHRW
      ISUMI=0
      CALL HDICE (ISUMI,NDICE,NSIDE,N)
      J=ISUMI
C     ADD UP NUMBER OF TIMES EACH SIDE COMES UP HEADS
      J=J&1
 40   ICNT(J)=ICNT(J)&1
C     THIS DO LOOP FINDS THE LARGEST VALUE OF ICNT(ISUM)
      DO 2001 ISUM=1,NDICE
      IF (KKK-ICNT(ISUM)) 2000,2001,2001
2000  KKK=ICNT(ISUM)
2001  CONTINUE
C     SET A SCALE FACTOR  FACT
      FACT=96./KKK
      S=0.0
      IP=0
      BIG=0.
      DO 930 IK=1,NDICE
      TEST=COEF(NDICE,IK,P)
      IF (TEST-BIG) 933,931,931
 931  BIG=TEST
 930  CONTINUE
 933  SCAL=96./(BIG*NTHRW)
C     FIND THE SMALLER SCALE FACTOR
      IF(SCAL-FACT) 151,151,152
 152   SCAL= FACT
 151  CONTINUE
C     MAKE THE SCALE FACTOR AN INTEGER IF IT IS GREATER THAN 1
      IF(SCAL-1.) 739,739,154
 154  SCAL=IFIX(SCAL)
C     WRITE DATA AND HEADER
 739  WRITE(3,740) NDICE,NSIDE,NTHRW
      WRITE(3,500)  SCAL
      WRITE(3,3555)IN
      DO 801 I=1,120
 801  ILINE(I)=DASH
      WRITE(3,802) ILINE
      WRITE(3,750)
      WRITE(3,6) IHEAD
      WRITE(3,703)
      WRITE(3,700)
      WRITE(3,701)
      GO TO (2302,704),ISW
 704  DO 150 ISUM=1,NDICE
      X=ICNT(ISUM)*SCAL
      JJ=ISUM-1
C     CHECK TO SEE THAT THERE ARE NO NEGATIVE VALUES
```

```
      IF (X) 900,910,210
900   WRITE (3,901)
      GO TO 2100
910   DO 250 I=1,96
250   LINEA(I)=BLANK
      GO TO 102
C     WRITE STARS UP TO AND INCLUDING THE SCALED VALUE OF X
210   DO 200 I=1,96
200   LINEA(I)=BLANK
      DO 101 I=1,X
101   LINEA(I)=STAR
102   CONTINUE
      S=COEF(NDICE,JJ,P)
      VALUE=S    *NTHRW
      IP=VALUE* SCAL
      DO 805 IK=1,96
805   LINET(IK)=BLANK
      LINET(IP)=O
      WRITE(3,160) JJ,DOT,LINEA,ICNT(ISUM)
      WRITE(3,161) LINET,VALUE
C     CHECK TO SEE IF THEORETICAL VALUES ARE DECREASING
      IF (VALUE-PVAL) 1111,149,149
1111  CONTINUE
C     STOP PRINTING AFTER VALUES DROP BELOW 0.01
      IF (VALUE-0.01) 3000,149,149
149   PVAL=VALUE
150   CONTINUE
3000  DO 803 I=1,120
803   ILINE(I)=DASH
      WRITE(3,802) ILINE
      GO TO 3
5     FORMAT(40A2)
6     FORMAT(' 'T38,40A2/)
160   FORMAT('0' I4,2X,A1,96A1,I5)
161   FORMAT('+'7X,96A1,8X,F9.3)
802   FORMAT(' '120A1/)
500   FORMAT (' A SCALE FACTOR OF'F5.2' IS COMPUTED'//)
700   FORMAT(' NUMBER OF TIMES A SIDE')
701   FORMAT(' APPEARS  IN  ONE  THROW',76X,'ACTUAL',3X,'THEORETICAL')
703   FORMAT(' ' 98X,'FREQUENCY OF OCCURRENCE')
740   FORMAT('1','A SET OF',I5,'   DICE,EACH DIE HAVING',I5,'  SIDES WAS
     X THROWN',I5,'  TIMES'//)
750   FORMAT('0'41X'HISTOGRAM OF FREQUENCY DISTRIBUTIONS'/)
901   FORMAT(' ERROR.  NEGATIVE NUMBER CALCULATED.  CALL EXIT.')
2202  FORMAT('0'I4,2X,A1,96A1,I5,F9.3)
2205  FORMAT('&'7X,96A1)
3555  FORMAT(' N='I5//)
      CALL EXIT
      END
```

```
C     PROGRAM EXPSM   A PROGRAM TO SIMULATE AN EXPONENTIAL DECAY.
      REAL NOTDK, NOTDC
      DIMENSION THEDC(41),THEDK( 41 ),IHEAD(40),LINE(120),
     XX(4),KDICE(41),IDK(41)
      DATA IDASH/'-'/
1000  READ(2,324) IHEAD
      NV=4
      CALL FREE(X,NV)
C     NDICE IS THE NUMBER OF DICE TO BE THROWN
      NDICE=X(1)
C     NSIDE IS THE NUMBER OF SIDES ON EACH DIE
      NSIDE=X(2)
C     N IS A STARTER FOR THE RANDOM GENERATION OF NUMBERS
      N=X(3)
C     NCYCL(OPTIONAL INPUT) IS THE NUMBER OF INTERVALS CALCULATED FOR
C     EACH PRINT
      NCYCL=X(4)
C     CHECK VALIDITY OF NCYCL
      IF (NV-3) 8,8,9
8     NCYCL=1
9     IF (NCYCL) 5,5,6
5     NCYCL=1
6     FACT=1.0/NSIDE
C     CHECK VALIDITY OF NSIDE
      IF(1000-NSIDE) 861,860,860
861   WRITE (3,863) IHEAD
      WRITE (3,862)NSIDE,NDICE,N,NCYCL
      GO TO 1000
860   THEDK(1)=0.0
      IDK(1)=0
      KIK=40
C     SET INITIAL VALUES
      THEDC(1)=NDICE
      ISW=1
      KDICE(1)=NDICE
      N=X(3)
      KK=0
      NOTDC=NDICE
      NOWDC=NDICE
      LPG=1
C     PRINT HEADER
      WRITE(3,320) LPG
      WRITE(3,450) NDICE,NSIDE
      WRITE(3,460) N ,NCYCL
      DO 10 I=1,120
10    LINE(I)=IDASH
      WRITE (3,321) LINE
      WRITE(3,322) IHEAD
      WRITE(3,323) LINE
      GO TO 99
470   WRITE(3,320) LPG
99    DO 500 I=1,KIK
      SUM=0.
      ISUM=0
      DO 499 K=1,NCYCL
C     CALCULATE VALUES
      CALL HDICE (NDK,NOWDC,NSIDE,N)
C     NDK IS ACTUAL NUMBER OF DECAYS IN THIS INTERVAL
C     NOWDC IS NUMBER OF DICE LEFT AFTER THIS DECAY
      NOWDC=NOWDC-NDK
      NOTDK=NOTDC*FACT
      SUM=SUM&NOTDK
      ISUM=ISUM&NDK
499   NOTDC=NOTDC-NOTDK
      II=I&1
C     THEDC IS THEORETICAL NUMBER OF DICE LEFT AFTER THIS DECAY
      THEDC(II)=NOTDC
      IDK(II)=ISUM
      KDICE(II)=NOWDC
C     THEDK IS THEORETICAL NUMBER OF DECAYS IN THIS INTERVAL
      THEDK(II)=SUM
C     CHECK TO SEE IF ALL DICE HAVE DECAYED
      IF (NOWDC) 200,200,500
500   CONTINUE
      GO TO 220
200   ITST=IODEV(I)
C     IF ODD NUMBER OF INTERVALS CALCULATED, GENERATE ANOTHER SET
C     IF EVEN NUMBER OF INTERVALS CALCULATED, CONTINUE
      GO TO (201,202),ITST
201   III=I&2
      IDK(III)=0
      KDICE(III)=0
      THEDK(III)=THEDC(II)*FACT
      THEDC(III)=THEDC(KK)-THEDK(II)
      I=II
202   IT=I/2
      ISW=2
      GO TO 551
220   IT=I/2
C     WRITE COLUMN HEADERS
551   WRITE(3,900)
      WRITE(3,901)
      WRITE(3,902)
      WRITE(3,910)
      DO 300 I=1,IT
      IO=I&IT
      J=(I+KK)*NCYCL
      JJ=(IO+KK)*NCYCL
      WRITE(3,310) KDICE(I),THEDC(I),KDICE(IO),THEDC(IO)
      WRITE (3,315) J,IDK(I&1),THEDK(I&1),JJ,IDK(IO&1),THEDK(IO&1)
300   CONTINUE
      KK=KIK&KK
C     IF NUMBER OF INTERVALS IS LESS THAN 200, CONTINUE
      IF (200-LPG*NCYCL*40)  698,698,699
698   ISW=2
699   GO TO (700,1000),ISW
C     REINITIALIZE VALUES
700   THEDC(1)=THEDC(KIK)
```

```
      KDICE(1)=KDICE(KIK)
      LPG=LPG&1
      GO TO 470
310   FORMAT(' 'T12,I5,T34,F8.2,T60,'I',T71,I5,T94,F8.2)
315   FORMAT(' 'I3,T22,I5,T46,F8.2,T60,'I',T65,I3,T83,I5,T105,F8.2)
320   FORMAT('1',T110,'PAGE'I4)
321   FORMAT('0',120A1)
322   FORMAT('0',T40,40A2)
323   FORMAT('0',120A1)
324   FORMAT(40A2)
450   FORMAT(' THERE ARE ORIGINALLY'I5'  DICE, EACH HAVING'I5'  SIDES'/)
460   FORMAT(' N='I5,'       PRINT FOR EVERY 'I5'  THROWS'//)
862   FORMAT(' INVALID DATA COMBINATION.  NSIDE IS GREATER THAN 1000.'/
     X' '4I5)
863   FORMAT('1' 40A2/)
900   FORMAT('0INTERVAL',T13,'ACTUAL',T24,'ACTUAL',T37,'THEOR.',T49,
     X'THEOR.',T60,'I',T63,'INTERVAL',T73,'ACTUAL'T84,'ACTUAL',T97,'THEO
     XR.'T109,'THEOR.')
901   FORMAT(' 'T14,'DICE',T25,'DICE'T37,'DICE'T 49,'DICE',T60,'I',T74,
     X'DICE',T85,'DICE'T97,'DICE',T109,'DICE')
902   FORMAT(' 'T14,'LEFT',T25,'DECAY',T37,'LEFT',T49,'DECAY',T60,'I',
     XT74,'LEFT',T85,'DECAY',T97,'LEFT',T109,'DECAY')
910   FORMAT(' 'T60,'I')
      CALL EXIT
      END
```

```
      FUNCTION FCTLN(N)
      DIMENSION FLG(100)
      IF(N-100)  5,300,300
    5 IF(ABS(FLG(3)-0.693) - 0.1) 100,100,10
   10 FLG(1) = 0.0
      FLG(2) = 0.0
      DO 20 I = 3,100
      Q = I-1
   20 FLG(I) = FLG(I-1) & ALOG(Q)
  100 FCTLN = FLG(N+1)
      RETURN
  300 FCTLN = 0.5*ALOG(2.0*N*3.14159) + FLOAT(N)*ALOG(N/2.718282) +
     X 0.5/(12.0*N - 1.0)
      RETURN
      END
```

```
       ENT        DICE
CNT    DC         -15
CNTR   DC         0
INSTR  SRA        14
USE    DC         0
DICE   DC         0
       LD         DICE
       STO        ADDN
       MDX   L    ADDN,2
       LD         ADDN
       STO        ADD
       LD    I    DICE
       STO        NSADD
       MDX   L    DICE,1
       LD    I    DICE
       STO        DICE
       LD    I    NSADD
       BSC        &
       EXIT
       LD    I    ADD
       M          M1
       SLT        16
       BSC   L    NEG,&Z
POS    STO   I    ADD
       STO        USE
       LD         CNT
       STO        CNTR
       LD         INSTR
       STO        BOX
LOOP   LD         USE
BOX    SRA        14
       SRT        1
       MDX   L    BOX,-1
       MDX   L    CNTR,1
       B          LOOP
       SLA        16
       SRT        1
       D     I    NSADD
       SLT        16
       A          ONE
       STO   I    DICE
       MDX   L    ADDN,1
       BSC   I    ADDN
NEG    A          BIG
       A          ONE
       MDX        POS
BIG    DC         32767
ONE    DC         1
M1     DC         899
ADDN   DC         0
ADD    DC         0
NSADD  DC         0
       END
```

```
      SUBROUTINE IDICE(ISUM,NDICE,NSIDE,N)
      DIMENSION
      DO 1000 I=1,NSIDE
 1000 ISUM(I)=0
      DO 30 I=1,NDICE
      CALL DICE (NSIDE,IUP,N)
      ISUM(IUP)=ISUM(IUP)&1
 30   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE HDICE(ISUM,NDICE,NSIDE,N)
      ISUM=0
      CALL DICE(NSIDE,NUP,N)
      DO 100 I=1,NDICE
      CALL DICE (NSIDE,IUP,N)
      IF (IUP-NUP) 100,110,100
 110  ISUM=ISUM&1
 100  CONTINUE
      RETURN
      END
```

BALLISTICS

Jeffrey Jalbert

Denison University
Granville, Ohio

INTRODUCTION

This computer-based exercise for the study of motion as predicted by Newtonian mechanics was designed as a pilot program to explore the logistics of incorporating the computer into an introductory-level course, to test student reactions, and to develop a library of appropriate instructional programs.

Due to the experimental nature of the course, a group of 12 to 15 volunteers were solicited from a class of 60 students, with the understanding that they would not receive direct credit for their participation in the program. The available facilities, viz., an IBM 1130 operated on a nontime-sharing basis, required the division of this pilot group into four subgroups of three to four students each. The necessity for this subdivision was borne out in practice; it took each group a total of two to four hours to finish working with one simulation program. This experience has led us to give serious consideration to a campus-wide time-shared system.

The students were familiarized with the operation of the computer in an introductory lecture which dealt with the procedures for entering programs, their operation and descriptions of the various types of possible output. In addition, a number of game-playing and picture-drawing programs were demonstrated to emphasize the capabilities of the computer. It was evident from student reaction that a handout sheet providing additional information would have been most helpful at this point.

Once the students were familiar with the basic concepts of computer operation, they began to participate in working sessions which lasted from one to two hours (a half-hour proved too short). Each session in which a new program was introduced was supplemented with a brief background lecture outlining the physics involved in the program, describing the techniques used to integrate the appropriate equations, and suggesting the relevance of this type of problem to physics and the world in general. Once familiar with the program, students were left completely on their own.

The pilot group represented a polyglot population, ranging from freshmen who might later pursue physics, to senior chemistry, pre-med and mathematics majors. Hence, the course emphasis was directed toward such questions as "What is physics?", and "How do physicists practice physics?", rather than toward calculus and the mathematical derivations involved.

Beyond the demands of assembling a minimal program library, providing write-ups, and determining the logistics of the course, a fundamental challenge in designing a computer-assisted course from scratch is the lack of available information from students, faculty, or other sources. Still, the course has been successful and has generated quite a bit of student enthusiasm.

BALIS, the program presented here, is the first one encountered by the students and, of the programs developed for this course, provides the most sophisticated output formats and instructional aids. Essentially, it presents a study of the motion predicted by Newtonian mechanics in a situation which is beyond the student's mathematical ability to analyze. Thus, he is led to the conclusion that, by supplementing his mathematical capabilities with a working knowledge of the computer, he can numerically attack any real problem he might encounter and obtain an answer which agrees with the results of an actual experiment.

STUDENT MANUAL

The program BALIS will enable you to explore the predictions that mechanics makes about the motion of a projectile by providing a reasonably accurate model of the physical conditions under which the motion occurs. To do this, both the retarding force due to air friction and the gravitational force should be included in the study.

Thus, the forces on the projectile in our model are

a. its weight, straight down

$$\vec{w} = m\vec{g}$$

b. the frictional force due to the impedance of the air. From general experience we know that this force is acting to slow the projectile down (i.e., it is opposite to the direction of the velocity). Also, this force increases proportionately with the speed of the projectile. (Stick your hand out the window of a moving car to test this.) As a first approximation, for speeds much less than the speed of sound, we will take the magnitude of this force to be proportional to the magnitude of the velocity $F \sim v$. Thus, we have the frictional force $\vec{F} \sim -\vec{v}$, the minus sign indicating that $\vec{F}$ is opposed to $\vec{v}$. This may be stated in the form of an equation:

$$\vec{F} = -k\vec{v}$$

where k represents some kind of frictional constant which depends on the nature of the medium and geometry of the projectile.

The net force acting on the projectile in our model is now:

$$\vec{F}_{net} = \vec{F} + \vec{w}$$

The acceleration of the projectile can be found from

$$m\vec{a} = \vec{F}_{net}$$

or

$$\vec{a} = -(k/m)\vec{v} + \vec{g}$$

$$= -\alpha\vec{v} + \vec{g}$$

For brevity we will call "α" the damping constant. Our model of the motion is given by this last equation. It now remains for us to "solve" this equation. That is, knowing the acceleration, find the velocity and position as a function of time given suitable initial conditions.

This looks like a rather formidable equation to integrate for the velocity. What does $\vec{v} = \vec{g}t - \alpha\int\vec{v}dt$ actually mean with $\vec{v}$ on both sides of the equation? Let's see how we might be able to study the problem in another way.

First, re-state the vector equations as two separate component equations:

$$a_x = -\alpha v_x$$

$$a_y = -\alpha v_y - g$$

with the y-axis vertically upward, as usual.

Each of these equations may now be integrated separately. The procedure is as follows: suppose that at time t_0 the velocity components are v_{0x} and v_{0y}, and that the x and y positions are x_0 and y_0. Now consider the time interval Δt just after t_0. During this interval nothing is constant, but for a short enough interval the accelerations are almost constant. Take these accelerations to be those at the start of the time interval. (I admit that this is an approximation!) Then, during the time interval Δt, both the x and y velocities change by an amount Δv_x and Δv_y where:

$$\Delta v_x = a_x \Delta t$$

$$\Delta v_y = a_y \Delta y$$

Remember that these are some sort of average changes in velocity over the entire time interval. We will assume that the average velocities over the entire time interval are

$$\bar{v}_x = v_{0x} + \Delta v_x/2$$

$$\bar{v}_y = v_{0y} + \Delta v_y/2$$

With these values for the average velocities, we are equipped to determine the changes in x and y position during the time interval

$$\Delta x = \bar{v}_x \Delta t$$

$$\Delta y = \bar{v}_y \Delta t$$

so that the new values for the positions and velocities at the end of the time interval are

$$x = x_0 + \Delta x \qquad v_x = v_{0x} + \Delta v_x$$

$$y = y_0 + \Delta y \qquad v_y = v_{0y} + \Delta v_y$$

We can now repeat this process for another time interval Δt, etc. This procedure can yield very good results when Δt is small;

it is very poor with a large Δt. (Why?)

The computer will ask you for the necessary input data: the initial velocity, the angle above the horizon $\theta_0 = \tan^{-1}(v_{0y}/v_{0x})$, and the damping constant. In addition, the computer will request a value for Δt. The above procedure is then implemented for several successive intervals Δt until 0.1 sec has passed. At this point, the results are stored and the computation continues until another 0.1-sec interval passes and the results are stored again, etc. It is possible to modify this 0.1-sec time interval for a longer or shorter duration of the computer study of the motion, with corresponding changes in the amount of computer time required. Simply turn switch zero on the console up and, when requested to do so, enter your own time interval (TI). This value will remain fixed until changed again by this procedure. You will find it necessary to use this option for input velocities much larger than 10m/sec.

What You Might Investigate by Using BALIS

These are only suggestions, think up some of your own too!

1. What is the best angle to shoot at? (Does this depend on the damping constant?)
2. What is the effect of changing the initial velocity?
3. What effect does damping have on your results?
4. What effect does the size of Δt have on the results?
5. Can you compare the results of the computer with analytic results for special cases?

One final comment: before you start, plan what you are going to do. Don't start off by randomly punching numbers into the machine.

TEACHER'S GUIDE

As it stands, the program BALIS is a completely self-contained package for studying the motion of a projectile under the simultaneous action of gravitational and viscous forces. It is semi-tutorial, in that it carries on a limited dialogue with the operator, and also carries out calculations using parameters obtained in that dialogue.

There are two basic parts to the package. The mainline program carries out the dialogue, requests data and performs the model calculations as specified in the Student Manual. These calculations are based on the data obtained during the dialogue. The mainline also handles the standard tabular output printing, i.e., the parameters that were used in the calculation and the detailed numerical results if this option is specified by the console entry switches.

The second part is a subroutine, PLOT2, that handles the graphical display. Originally this was an IBM supplied package, but it has been modified so as to be more compatible with the present usage. Normally, the graphs that are produced by this subroutine will be scaled to fill an entire page (120 columns), and to occupy a varying number of lines (maximum of 100), depending upon how much input data is given. Either the vertical (Y-Scale) or horizontal (X-Scale), or both can be retained from one calculation to the next by raising the appropriate switches. For best results, a continuous tape should be used on the line printer. This eliminates unsightly skipping right in the middle of a graph.

It is quite easy to obtain vast amounts of output from this program. With a choice of the two output methods available, it is felt that for most purposes the graphical display is sufficient. If necessary, a given calculation can be repeated to obtain the numerical tabulation.

The option of retaining either or both the X-Scale and Y-Scale factors constant from one calculation to the next allows you to visually compare one trajectory to another that has different parameters for the model. The impact of this kind of comparison can be quite substantial. If this procedure would lead to an error in the plotting (graph goes off scale), the computer will so report and then return to the main program and continue just as if the graph had been produced.

For those students who are interested, you can show them how to integrate the model formula $\vec{a} = -\alpha\vec{v} + \vec{g}$ analytically. (See, e.g., Symon's Mechanics). The results are

$$x = (v_{0x}/\alpha)\ (1-e^{-\alpha t})$$

$$y = ((g/\alpha^2)+(v_{0y}/\alpha))(1-e^{-\alpha t}) - (gt/\alpha)$$

Calculations using these formulae can be compared with the numerical tabulations that can be obtained from BALIS. (The program uses $g = 10\text{m/sec}^2$ for simplicity.) Alternatively, one could check out various levels of approximation to these formulae.

The most frequent questions that students ask include: "What is the damping constant α?"; "What value does it have?". These students might be referred to a discussion of Stoke's law for the resistance to a sphere moving through a fluid, where they can get order-of-magnitude estimates that could be reasonable in a real atmosphere.

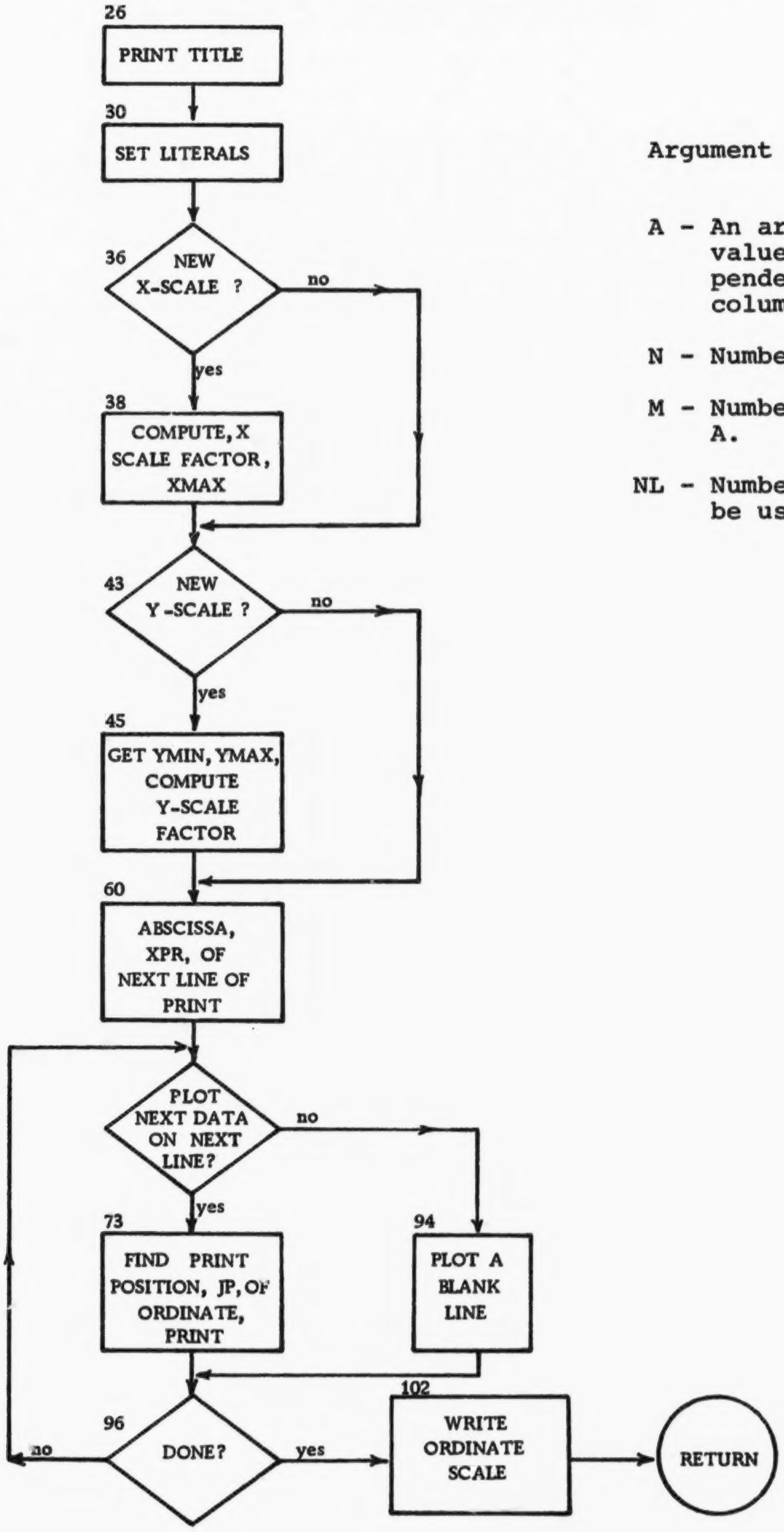
The programming language used to write these programs is FORTRAN IV as adapted for the IBM 1130 computer. Most of the statements are just standard FORTRAN and should not have to be modified for use on any other machine. There are, however, some things that may have to be changed. The first is the method of interrogating the console entry switches. This is done in 1130 FORTRAN by the statement CALL DATSW(I,J). This statement causes console switch I to be interrogated. If this switch is on, J is set to 1. If the switch is off, J is set to 2.

The second problem is logical device specification. This is handled in read and write commands by inserting (I,J) just after read or write. I specifies the logical unit and J specifies the format statement number. The logical number for the line printer is specified in the first statement of the subroutines "MX = 5" and may be changed there. In all other cases the use of "1" in the listing implies the console typewriter and "5" the line printer in 1130 FORTRAN.

A third problem that may arise is the method used by the plot subroutine. The characters that will be used are a blank (no mark) repeated to obtain appropriate distance along the line and a star for the point. The integer equivalent of these for the 1130 is 16448 and 23616, respectively. If your machine uses a different literal code than that which is used in the statements IBLNK= and IANG= (on cards 30 and 31, respectively) these statements should appropriately modified.

The flow charts on the following pages illustrate the operation of BALIS in some detail. The numbers on the flow charts indicate the card numbers at which various processes begin. A sample of the output is also shown.

Flow Chart for Subroutine PLOT2(A,N,M,NL) of BALIS


26
PRINT TITLE
30
SET LITERALS
36
NEW X-SCALE ?
no
yes
38
COMPUTE, X SCALE FACTOR, XMAX
43
NEW Y-SCALE ?
no
yes
45
GET YMIN, YMAX, COMPUTE Y-SCALE FACTOR
60
ABSCISSA, XPR, OF NEXT LINE OF PRINT
PLOT NEXT DATA ON NEXT LINE?
no
yes
73
FIND PRINT POSITION, JP, OF ORDINATE, PRINT
94
PLOT A BLANK LINE
96
DONE?
no
yes
102
WRITE ORDINATE SCALE
RETURN

Argument List:

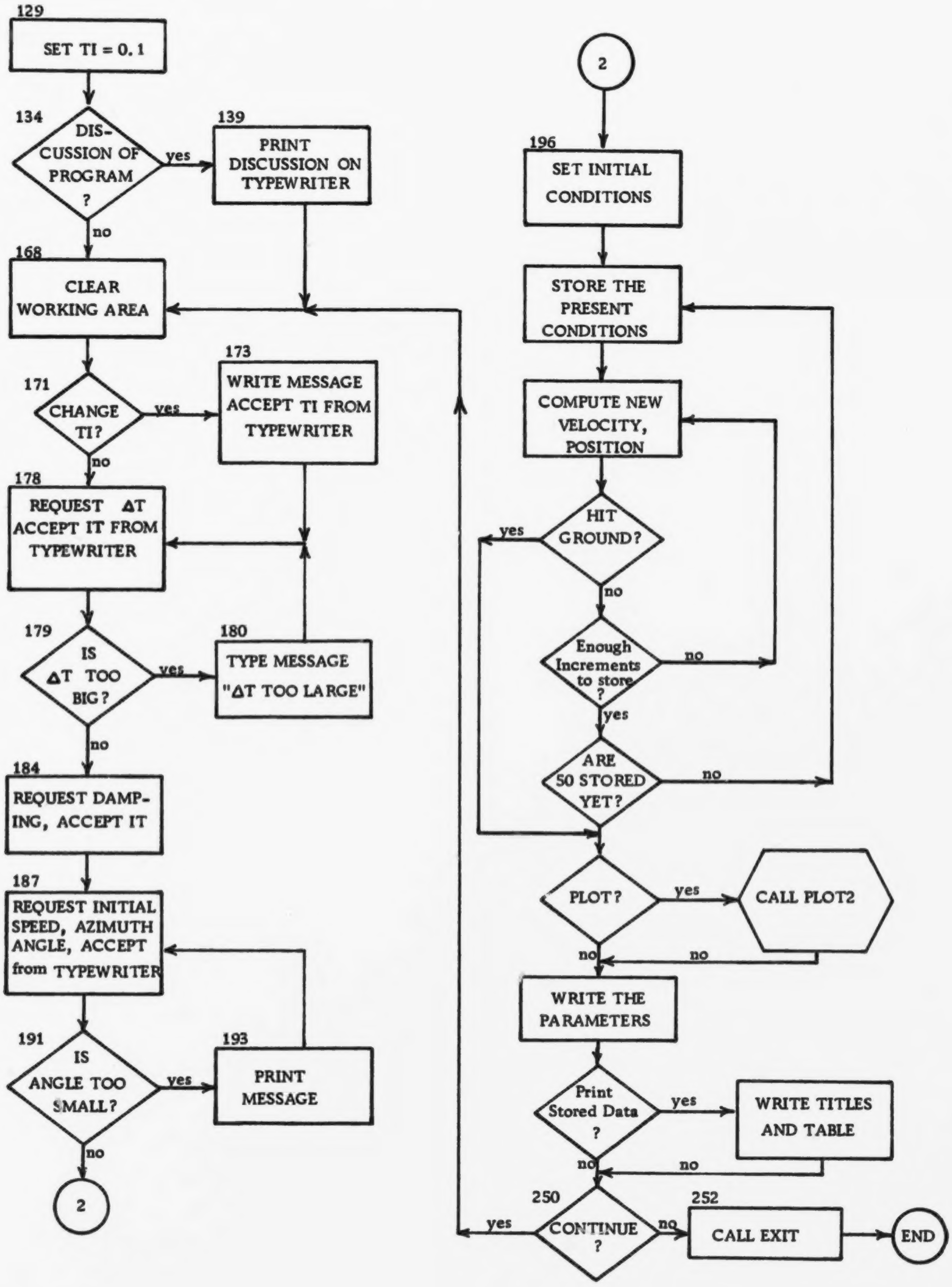
A - An array containing values of the independent variable in column 2.

N - Number of rows of A.

M - Number of columns of A.

NL - Number of lines to be used in display.

Flow Chart for BALIS, Mainline Program


129
SET TI = 0.1
134
DIS-CUSSION OF PROGRAM ?
yes
139
PRINT DISCUSSION ON TYPEWRITER
no
168
CLEAR WORKING AREA
171
CHANGE TI?
yes
173
WRITE MESSAGE ACCEPT TI FROM TYPEWRITER
no
178
REQUEST ΔT ACCEPT IT FROM TYPEWRITER
179
IS ΔT TOO BIG?
yes
180
TYPE MESSAGE "ΔT TOO LARGE"
no
184
REQUEST DAMPING, ACCEPT IT
187
REQUEST INITIAL SPEED, AZIMUTH ANGLE, ACCEPT from TYPEWRITER
191
IS ANGLE TOO SMALL?
yes
193
PRINT MESSAGE
no
2
2
196
SET INITIAL CONDITIONS
STORE THE PRESENT CONDITIONS
COMPUTE NEW VELOCITY, POSITION
HIT GROUND?
yes
no
Enough Increments to store ?
no
yes
ARE 50 STORED YET?
no
PLOT?
yes
CALL PLOT2
no
no
WRITE THE PARAMETERS
Print Stored Data ?
yes
WRITE TITLES AND TABLE
no
no
250
CONTINUE ?
yes
no
252
CALL EXIT
END

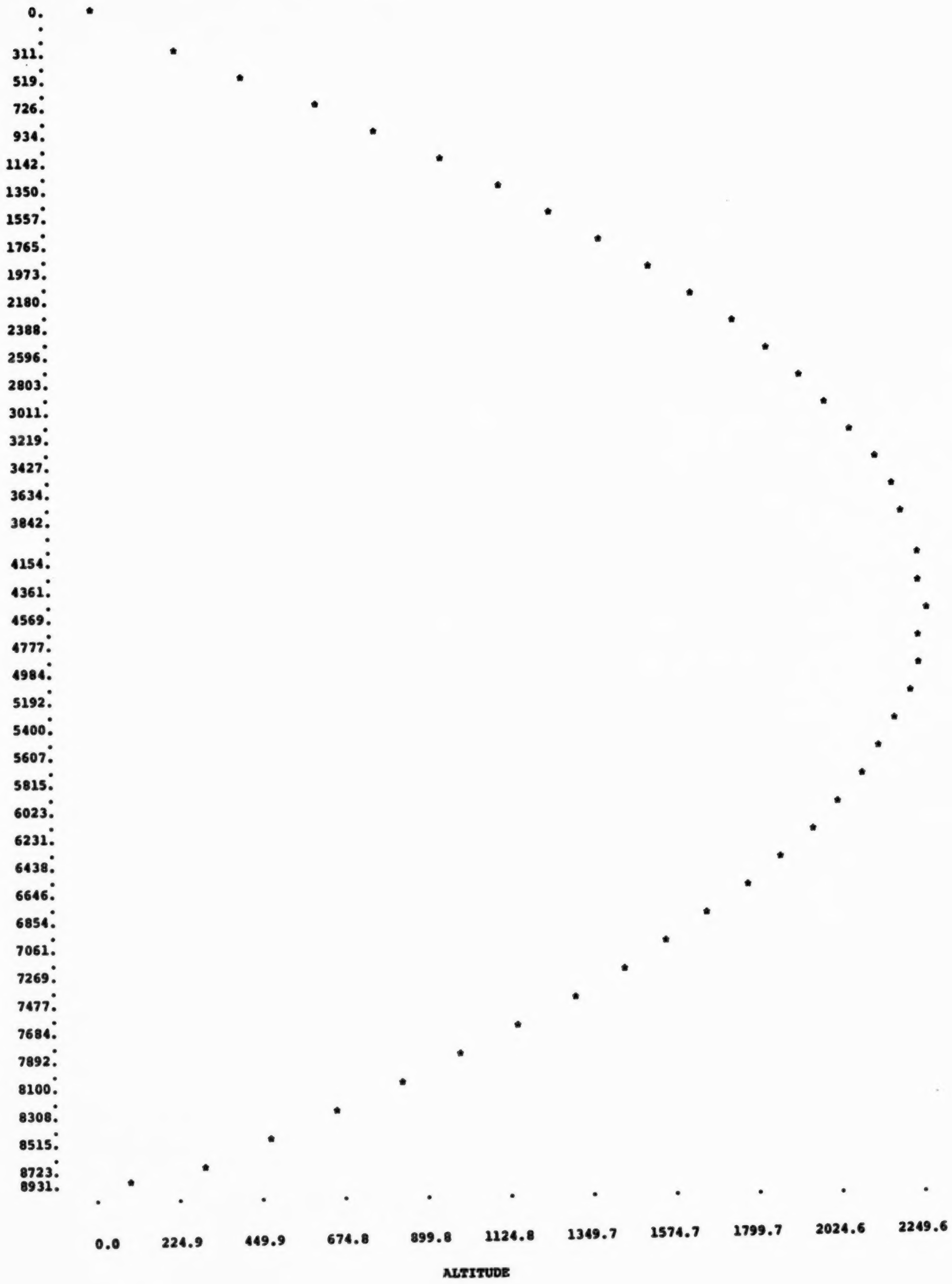
```
// JOB
// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE PLOT2(A,N,M,NL)
C.......................................................................
C
C     SUBROUTINE TO PLOT
C
C.......................................................................
      DIMENSION OUT(101),YPR(11),A(1)
      MX=5
    1 FORMAT(////////,46X,'...........................',/,46X,'.',25X,
     1'.',/,46X,'.   GRAPH OF TRAJECTORY   .',/,46X,'.    ALTITUDE VS RA
     2NGE    .',/,46X,'.',25X,'.',/,46X,'...........................',
     3//)
    2 FORMAT(1X,F11.0,5X,101A1)
    3 FORMAT(11X,1H.)
    7 FORMAT(    16X,101H.         .         .         .         .
     1 .           .         .         .         .         .)
    8 FORMAT(//,9X,11F10.1)
      NLL=NL
C
C        PRINT TITLE
C
      WRITE(MX,1)
C
C     SET PRINT LIT'S
C
      IBLNK=16448
      IANG=23616
      LN=NLL/2
C
C     FIND SCALE FOR BASE VARIABLE  CHECK TO SEE IF IT IS TO BE CHANGED
C
      CALL DATSW(4,J)
      GO TO(21,20),J
   20 XSCAL=(A(LN)-A(1))/FLOAT(NLL)
      XMAX=A(LN)
C
C        FIND SCALE FOR CROSS-VARIABLES CHECK TO SEE IF TO BE CHANGED
C
   21 CALL DATSW(5,J)
      GO TO(41,25),J
   25 M1=N+1
      M2=M*N
      YMIN=A(M1)
      YMAX=YMIN
      DO 40 J=M1,M2
      IF(A(J)-YMIN) 28,26,26
   26 IF(A(J)-YMAX) 40,40,30
   28 YMIN=A(J)
      GO TO 40
   30 YMAX=A(J)
   40 CONTINUE
      YSCAL=(YMAX-YMIN)/100.0
C
C         FIND BASE VARIABLE PRINT POSITION
C
   41 XB=A(1)
      L=1
      MYX = M-1
      I=1
   45 F=FLOAT(I-1)
      XPR=XB+F*XSCAL
      IF(A(L)-XPR)46,50,70
   46 IF(A(L+1)-XPR-.1)47,47,50
   47 L=L+1
      IF(L-LN)46,50,50
C
C         FIND CROSS-VARIABLES
C
   50 IF(A(L)-XMAX)51,51,52
   52 WRITE(1,53)
   53 FORMAT('X-SCALE IS TOO SMALL FOR PRESENT DATA')
      GO TO 88
   51 DO 55 IX=1,101
   55 OUT(IX)=IBLNK
      DO 60 J=1,MYX
      LL=L+J*N
      JP=((A(LL)-YMIN)/YSCAL)+1.0
      IF(A(LL)-YMAX) 56,56,57
   57 WRITE(1,58)
   58 FORMAT('Y-SCALE IS TOO SMALL FOR PRESENT DATA')
      GO TO 88
   56 OUT(JP)=IANG
   60 CONTINUE
C
C         PRINT LINE AND CLEAR, OR SKIP
C
      WRITE(MX,2)XPR,(OUT(IZ),IZ=1,101)
      L=L+1
      GO TO 80
   70 WRITE(MX,3)
   80 I=I+1
      IF(L-LN)45,84,86
   84 XPR=A(LN)
      GO TO 50
C
C         PRINT CROSS-VARIABLES NUMBERS
C
   86 WRITE(MX,7)
      YPR(1)=YMIN
      DO 90 KN=1,9
   90 YPR(KN+1)=YPR(KN)+YSCAL*10.0
      YPR(11)=YMAX
      WRITE(MX,8)(YPR(IP),IP=1,11)
```

```
C
C     PRINT TITLE
C
      WRITE(5,87)
   87 FORMAT(//58X,'ALTITUDE')
   88 RETURN
      END
// DUP
*STORE      WS  UA  PLOT2
// FOR
*NAME BALIS
*IOCS (CARD,TYPEWRITER, KEYBOARD, 1403PRINTER)
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
C.......................................................................
C
C        PROGRAM TO STUDY THE MOTION OF A PROJECTILE UNDER
C        SIMULTANEOUS ACTION OF GRAVITATIONAL AND VISCOUS FORCES
C
C.......................................................................
      DIMENSION SAV(50,3)
      TI=.1
C
C     INTERACTION SECTION
C
      WRITE(1,40)
   40 FORMAT('DO YOU WISH A DISCUSSION OF THIS PROGRAM  SWITCH 15 UP = Y
     1ES,  PUSH START')
      PAUSE
      CALL DATSW(15,J)
      GO TO(41,1),J
   41 WRITE(1,28)
   28 FORMAT('THIS PROGRAM ALLOWS YOU TO STUDY THE MOTION OF A BALLISTIC
     1 PARTICLE (E.G. A BULLET) AS PREDICTED BY NEWTONS'/'LAWS OF MOTION
     2. TO MAKE THE SITUATION MORE REALISTIC PROVISION IS MADE TO INCLUD
     3E A RETARDING FORCE PROPORTIONAL'/'TO THE VELOCITY (PROPORTIONATEL
     4Y CONSTANT = K). YOU WILL BE ABLE TO CHANGE THE PARAMETERS (INITIA
     5L CONDITIONS,')
      WRITE(1,29
   29 FORMAT ('RETARDING FORCE) THAT ENTER INTO THE THEORY AND SEE THE E
     1FFECTS THAT THESE HAVE ON THE PREDICTED MOTION. THE'/'FUNCTION OF 
     2THE COMPUTER IS TO RELIEVE YOU OF THE TEDIUM INVOLVED IN PERFORMIN
     3G THE CALCULATIONS. IT MUST HAVE'/'ALL OF THE DATA THAT YOU YOURSE
     4LF WOULD NEED TO DO THE SAME CALCULATION. IT WILL REQUEST THESE AS
     5 NECESSARY.')
      WRITE(1,30)
   30 FORMAT(/'CONVENTIONS USED ARE'/,5X,'1. MKS UNITS'/,5X,'2. TIME STE
     1P = INTERVAL OVER WHICH ACCELERATION ASSUMED CONSTANT'/,5X,'3. DAM
     2PING CONSTANT = K/MASS'/,5X,'4. AZIMUTH ANGLE = ANGLE ABOVE HORIZO
     3N'/,5X,'5. EARTH IS FLAT')
      WRITE(1,31)
   31 FORMAT('IN ADDITION YOU MAY SET THE CONSOLE SWITCHES TO MODIFY THE
     1 FLOW OF THE PROGRAM. THE FOLLOWING RESULTS ARE OBTAINED...')
      WRITE(1,24)
   24 FORMAT(5X,'SWITCH 2 ON TO GRAPH'/,
     15X,'SWITCH 3 ON FOR PRINTOUT'/,
     25X,'SWITCH 4 ON TO RETAIN PREVIOUS X-SCALE'/,
     35X,'SWITCH 5 ON TO RETAIN PREVIOUS Y-SCALE'/)
      WRITE(1,36
   36 FORMAT('SWITCHES 4 AND 5 MUST BE OFF ON THE FIRST PASS.'//)
    1 DO 10 I=1,50
      SAV(I,1)=0.
   10 SAV(I,2)=0.
      CALL DATSW(0,I)
      GO TO(14,15),I
   14 WRITE(1,16)
   16 FORMAT('ENTER TI, MASTER')
      READ(6,5)TI
   15 WRITE(1,2)
    2 FORMAT('ENTER TIME STEP AS A 5 PLACE NUMBER')
      READ(6,5)DELT
      IF(DELT-0.1)37,37,39
   39 WRITE(1,38)
   38 FORMAT('YOUR TIME STEP IS TOO LARGE, WE WILL TRY AGAIN')
      GO TO 15
   37 WRITE(1,3)
    3 FORMAT('ENTER DAMPING CONSTANT (5 PLACES)')
      READ(6,5)DAMC
   34 WRITE(1,4)
    4 FORMAT( 'ENTER INITIAL SPEED AND AZIMUTH ANGLE (IN DEGREES), PLEAS
     1E USE 5 PLACES FOR EACH')
    5 FORMAT(2F5.3)
      READ(6,5)SPD,THETA
      IF(THETA-0.0000)35,32,35
   32 WRITE(1,33)
   33 FORMAT('YOUR ANGLE IS 0. THE PARTICLE HIT THE GROUND IMMEDIATELY A
     1FTER FIRING. TRY AGAIN.')
      GO TO 34
   35 CONTINUE
C
C         SET INITIAL CONDITIONS
C
      VXI=SPD*COS(THETA/57.3)
      VYI=SPD*SIN(THETA/57.3)
      VX=VXI
      VY=VYI
      T=0.
      TP=0.
      I=0
      X=0.
      Y=0.
C
C         BEGIN ACTUAL CALCULATIONS
C
    6 I=I+1
      SAV(I,1)=X
      SAV(I,2)=Y
      SAV(I,3)=TP
      TP=TP+TI
      IF(I-50)7,27,27
    7 T=T+DELT
      DELVY=-(DAMC*VY+10.) *DELT
      DELVX=-DAMC*VX*DELT
      X=X+(VX+DELVX/2.)*DELT
      Y=Y+(VY+DELVY/2.)*DELT
      IF(Y+.1)27,27,71
   71 VX=VX+DELVX
      VY=VY+DELVY
      IF(T-TP)7,6,6
C
C         GENERAL HOUSEKEEPING
C
   27 CALL DATSW(2,J)
      GO TO(8,23),J
    8 CALL PLOT2(SAV,50,2,2*I)
      WRITE(5,20)DAMC,SPD,THETA
   20 FORMAT(///' INITIAL CONDITIONS'/' DAMPING CONSTANT = ',F5.3,2X,
     1'INITIAL SPEED = ',F6.1,2X,'AZIMUTH ANGLE = ',F5.1,' DEGREES')
   23 CALL DATSW(3,J)
      GO TO(25,26),J
   25 WRITE(5,21)
   21 FORMAT(///'  TABLE OF HORIZONTAL AND VERTICAL'/' DISPLACEMENTS AS
     1A FUNCTION OF TIME')
      WRITE(5,17)
   17 FORMAT(//' TIME',14X,'X',12X,'Y')
      DO 19 J=1,I
   19 WRITE(5,18)SAV(J,3),SAV(J,1),SAV(J,2)
   18 FORMAT(1H0,F5.2,8X,f(.3,3X,F9.3)
   26 WRITE(1,9)
    9 FORMAT('SWITCH ONE ON TO EXIT')
      PAUSE
      CALL DATSW(1,I)
      GO TO(11,1),I
   11 CALL EXIT
      END
```

```
INITIAL CONDITIONS
DAMPING CONSTANT = 0.000  INITIAL SPEED = 3000.0  AZIMUTH ANGLE =  45.0 DEGREES


 TABLE OF HORIZONTAL AND VERTICAL
DISPLACEMENTS AS A FUNCTION OF TIME


TIME                X           Y
 0.00             0.000       0.000
 0.10           233.358     227.281
 0.20           445.502     423.401
 0.30           657.646     609.520
 0.39           869.790     785.639
 0.49          1081.933     951.758
 0.59          1294.077    1107.877
 0.69          1506.220    1253.996
 0.79          1718.364    1390.114
 0.89          1930.508    1516.233
 0.99          2142.651    1632.352
 1.09          2354.795    1738.470
 1.19          2566.938    1834.589
 1.29          2779.082    1920.708
 1.39          2991.226    1996.826
 1.49          3203.369    2062.945
 1.59          3415.513    2119.062
 1.69          3627.656    2165.179
 1.79          3839.800    2201.296
 1.89          4051.943    2227.414
```

1.99	4264.083	2243.531
2.09	4476.222	2249.648
2.19	4688.361	2245.765
2.29	4900.500	2231.882
2.39	5112.638	2208.000
2.49	5324.777	2174.117
2.59	5536.916	2130.234
2.69	5749.054	2076.351
2.79	5961.193	2012.469
2.89	6173.332	1938.588
2.99	6385.470	1854.706
3.09	6597.609	1760.825
3.19	6809.748	1656.944
3.29	7021.886	1543.062
3.39	7234.025	1419.181
3.49	7446.164	1285.300
3.59	7658.302	1141.418
3.69	7870.441	987.537
3.79	8082.580	823.656
3.89	8294.714	649.775
3.99	8506.843	465.894
4.09	8718.972	272.014
4.19	8931.101	68.133

GRAPH OF TRAJECTORY
ALTITUDE VS RANGE



OPENING THE OTHER END

Ronald Blum

Commission on College Physics
University of Maryland
College Park, Maryland

INTRODUCTION

One often hears of the desirability of "open-ended" problems in the classroom and laboratory; problems in which the student begins with a relatively simple situation (for example, a falling body) and then proceeds to add further refinements and nuances. Thus, he gradually expands his understanding until he is coming to grips with problems of general significance, convincing realism and wide application. The computer promises to be the tool *par excellence* for attaining this goal and "opening the other end" to the appreciation of physicists and their students.

As an example of this educational use of the computer, we have chosen to treat the problem of a ballistic projectile moving through a resistive fluid. Thus, this paper may be considered a natural extension of the ballistic problem presented in the low-velocity limit by Jalbert.* Although intended for a more sophisticated audience, such as a class in advanced undergraduate mechanics would afford, this material should be accessible to students with some elementary computing experience and an understanding of the concepts of differential equations, integrals and the meaning of Newton's second law, $F = Ma$. A student capable of integrating the second law of motion in its differential form for a projectile in a vacuum should be capable of understanding the following material, although the details will require some study. As for the numerical techniques involved--i.e., the Trapezoidal Rule and the improved Euler method--these can be readily appreciated by anyone with an understanding of the geometrical significance of an integral, i.e., the fundamental theorem of calculus.

After first discussing the nature of fluid friction and the significance of the Reynolds number and drag coefficient, we then consider three cases in order of increasing complexity: (1) horizontal motion over a frictionless surface, (2) vertical motion in the atmosphere, and (3) the reentry problem as an example of two-dimensional motion in the atmosphere. In the horizontal case, the problem is fairly straightforward, and it can be solved analytically for the case of constant drag. However, it does serve as a useful vehicle for introducing the drag as a tabulated function, derived from experimental data, and evaluated by means of a "table look-up" in the course of the numerical integration of the equation of motion. It is shown that the realistic drag model causes the motion to stop (at $t = \infty$) in a much shorter distance than it would have if one had assumed either Stokes drag (low-velocity) or Newtonian drag (high-velocity). This somewhat surprising result affords some useful insights into the nature of the high- and low-velocity resistive forces.

---

*"Ballistics," Jeffrey Jalbert, in *Computer-Based Physics: An Anthology*, R. Blum, et al, published by the Commission on College Physics (1969).

The vertical case introduces gravity and a varying density into the equations of motion, precluding any simple analytic solution even in the case of constant drag. At this point we have cast loose from conventional physics instruction and must rely on the computer. Furthermore, it also becomes necessary to replace the second-order equation of motion by two first-order equations of motion, treating the velocity as a separate variable. This technique, of great value in the theory of differential equations as well as in numerical analysis, is a precursor of the Hamiltonian formulation of mechanics.

The reentry problem is the two-dimensional synthesis of the preceding cases. However, although it represents a significant increase of our capacity to handle real problems, it does not present a corresponding increase in difficulty over the preceding two cases. Although, for a time-dependent solution, we should require four first-order equations, if time is eliminated by the use of altitude y as the independent variable, then the horizontal range, x, need not be computed simultaneously with the velocity components $v_x = u(y)$ and $v_y = v(y)$. Hence, the resulting system of equations represents no more difficulty than the time-dependent solution of the vertical case. This is, in fact, the essence of open-endedness: by a gradual process to make one's way to higher and higher levels of analytic power and realism without correspondingly increasing the complexity of the conceptual base.

It should be noted that none of these programs has yet been offered to a physics class. We have instead availed ourselves of editorial license to include these untried programs in this work. Hence, it is not possible to make any definitive recommendations concerning classroom usage. However, it might be suggested that the general theory and the horizontal case would be suitable for a single lecture, given a prior understanding of the basic mechanics involved. This could then be followed up by a homework-and-laboratory assignment to program and compute the vertical case, the reentry problem, or both. It is recognized that the entire unit represents a rather stiff dose of open-endedness; one may find that only the better students would be genuinely stimulated to go on to the reentry problem. Complete programs will be found in the Teacher's Guide at the end of this paper. They are written in BASIC, reflecting our view that time-shared systems are most appropriate to the physics classroom.

General Theory

In your work thus far you have studied the motion of a freely falling point mass, which illustrated the kinematic relations between acceleration, velocity and displacement. The next step, to analyze the two-dimensional motion of a ballistic projectile (neglecting frictional effects), began with Newton's second law of motion, $F = Ma$. By applying the concept of a vector to the dynamics of such a particle one can derive its time-dependent behavior under a constant gravitational acceleration, g, and show that it is quadratic and linear in time in its vertical (y) and horizontal (x) coordinates, respectively. The result is a trajectory which is parabolic, of the form

$$y = y_m + a(x-x_m)^2 \tag{1}$$

where

$$\begin{aligned} x_m &= x_0 + u_0 v_0/g \qquad (2) \\ y_m &= y_0 + v_0^2/2g \\ a &= -g/2u_0^2 \end{aligned}$$

the zero-subscripted quantities ($_0$) referring to initial values of displacement and velocity, $\underline{v} = u\underline{i} + v\underline{j}$.

However, terrestrial projectiles do not move in a vacuum, they move through the air, hence there is frictional resistance to their motion. The exact nature of this resistance is quite complicated, and varies according to the speed of the projectile and the properties of the fluid (gas or liquid) through which it flows--the atmosphere in this case. Osborne Reynolds (1842-1912) showed that a dimensionless parameter, the Reynolds number, N_R, could be defined which characterizes the flow of fluid around a body moving through it, and hence also determines the resistance which the body encounters as it forces its way through the fluid. The Reynolds number is defined as

$$N_R = \frac{\rho V L}{\eta} \tag{3}$$

where ρ and η are the density and the viscosity coefficient of the fluid, V is the flow speed relative to the particular solid body exposed to the fluid and L is a length "characteristic" of that solid body. For flow past a sphere, L is the diameter $2R$ of the sphere. In the following analysis, we shall assume our projectile to be spherical.

For an object moving with velocity $\underline{v}$ through a stationary fluid, the relative flow speed is $V = |\underline{v}|$. Experiment and theory show that the resistive force can be conveniently expressed by

$$\underline{f} = -6\pi\eta R\underline{v}\left(\frac{C_D N_R}{24}\right) \tag{4}$$

where C_D is a dimensionless parameter that can be evaluated experimentally and is called the drag coefficient. Although C_D is a function of various variables, mainly the Reynolds number N_R, it varies only slowly over a fairly large range of values of $N_R > 100$. Hence it is often useful to select a constant average value of the drag coefficient C_D and, as you can easily show with the help of Eq. (3), to convert Eq. (4) to the form known as Newton's resistance law:

$$\underline{f} = -\tfrac{1}{2}C_D\rho V A\underline{v} \tag{5}$$

where A is the cross-sectional area of the sphere, πR^2.

You have probably already studied the effect of Stokes'-law resistance on projectile motion, in which $\underline{f} = -k\underline{v}$, where k is a constant.* George Stokes (1819-1903) showed in 1845 that

$$\underline{f} = -6\pi\eta R\underline{v} \tag{6}$$

for a sphere moving so slowly that the flow of fluid around the sphere is smooth (laminar) and nonturbulent. In this case the parametric equations of a projectile trajectory take the form ($\alpha = k/M$):

$$x = \frac{u_0}{\alpha}(1-e^{-\alpha t}) + x_0 \tag{7}$$

$$y = (g/\alpha^2 + v_0/\alpha)(1-e^{-\alpha t}) - gt/\alpha + y_0$$

Problem 1. Show that in the limit $\alpha \to 0$ Eqs. (7) reduce to the usual equations of projectile motion in a vacuum. (Hint: expand $e^{-\alpha t}$ in the first few terms of a Taylor's series.)

If $C_D = C_D(N_R)$, then at low speeds Eq. (4) must reduce to Stokes' law; i.e.,

$$\lim_{N_R \to 0} C_D = 24/N_R \tag{8}$$

whereas at high speeds and large N_R, C_D = constant, approximately. Figure 1 illustrates the actual dependence of $C_D(N_R)$ for a sphere. The curve varies depending upon the shape and orientation of the projectile (determining $C_D(N_R)$ for various configurations is a major problem in aerodynamics); the sudden dip in the curve at $N = 200{,}000$ indicates the critical Reynolds number, where the onset of turbulent flow is manifested by a sudden drop in the drag coefficient. For large Reynolds numbers we will simply assume

$$C_D = 0.5 \quad \text{for} \quad 1{,}000 \le N_R \tag{9}$$

---

*Jalbert, Jeffrey, op. cit.

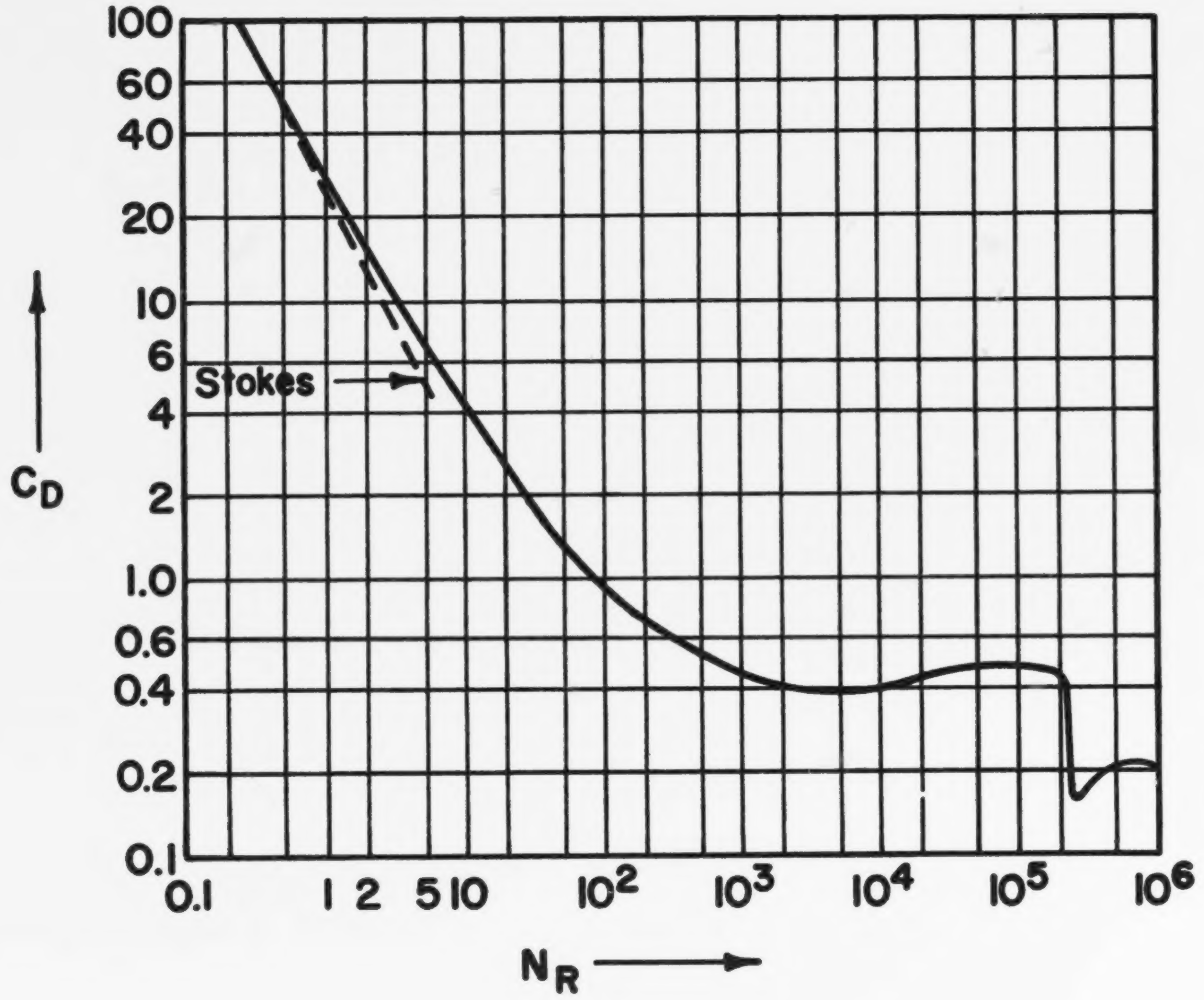


Figure 1. Dependence of drag coefficient, C_D, on Reynolds number, N_R.

Problem 2. Show that $C_D = 0.5$ combined with Eq. (5) implies that the resistive force on the sphere can be thought of as due to an average gain of speed of $V/4$ parallel to the projectile's path by fluid particles contained within the volume through which the projectile travels. (Hint: first show that ρVA grams of fluid per second are "swept out" by the projectile, then find the average rate of change of momentum of this fluid.)

Although we find the equations of projectile motion to be soluble in the limit of Stokes' resistance (Eqs. (7)), the nonlinear form of Newton's resistance, Eq. (5), makes life much more difficult. Furthermore, for a Reynolds number $N_R = 10{,}000$, for example, the Newtonian resistance ($C_D N_R/24 = 200$) is approximately 200 times greater than that which would be predicted by Stokes' law. To deal with the projectile problem most generally, one must, therefore, resort to numerical computations, continually computing new values of C_D from a table as the speed of the projectile changes. Another significant factor, when the fluid is the atmosphere, is the marked change in density with altitude. In the following pages, we shall illustrate the nature of this problem and its numerical solutions by considering three cases: (1) horizontal motion on a frictionless surface, (2) vertical motion through the atmosphere, and (3) the motion (two-dimensional) of a reentry body as it returns to Earth from space.

Numerical Integration and Quadrature

The numerical technique we shall use to integrate the equations of motion is the improved Euler method, a second-order method in which errors are proportional to $(\Delta t)^3$, Δt being the integration-step. This method is a member of a class of higher-order techniques for integration known as Runge-Kutta methods. A fourth-order Runge-Kutta method* enjoys very widespread use, and is to be found in the program library of every well-equipped computing center.

Consider a curve $x(t)$ such that

$$dx/dt = f(t,x) \tag{10}$$

Given some value $x_n = x(t_n)$ we could find

$$x_{n+1} = x_n + \int_{t_n}^{t_{n+1}} f(t,x)\,dt$$

if we had some way of evaluating the integral without having prior knowledge of $x(t)$. We do this by computing the approximate average slope $\bar{f}$ over the interval and setting

$$x_{n+1} = x_n + \bar{f}\Delta t \tag{11}$$

---

*Vierling, Anton F., "Harmonic Motion," op. cit.

where $\Delta t = t_{n+1} - t_n$.

The obvious way to do this is to set

$$f = \tfrac{1}{2}[f(t_n,x_n) + f(t_{n+1},x_{n+1})] \tag{12}$$

However, x_{n+1} is what we are trying to determine; we cannot tolerate its presence on both sides of Eq. (11), and if $f(t,x)$ were simple enough to allow us to solve Eqs. (11) and (12) simultaneously for x_{n+1} we should probably not have had to resort to numerical methods in the first place. Therefore, we must find some way to approximate x_{n+1} on the right-hand side of Eq. (12). This we do by means of a first-order prediction

$$x_{n+1} \doteq X = x_n + f(t_n,x_n)\Delta t \tag{13}$$

which we then use to find

$$f(t_{n+1},x_{n+1}) \doteq f(t_{n+1},X)$$

$$\therefore \quad x_{n+1} = x_n + \tfrac{1}{2}\Delta t[f(t_n,x_n) + f(t_{n+1},X)] \tag{14}$$

This process may be pictured graphically (see Figure 2) as follows: draw the tangent to the curve through (t_n,x_n) with slope $f(t_n,x_n)$. This line, τ_1, determines X, the first-order approximation in Eq. (13). Now, if we pass a line τ_2 with slope $f(t_{n+1},X)$ through this point it should be very nearly parallel to the actual tangent to the curve $f(t_{n+1},x(t_{n+1}))$; hence, the average slope over the interval is approximately $\overline{f} = [f(t_n,x_n) + f(t_{n+1},X)]/2$ as indicated in Eq. (14), and illustrated by the dashed line, τ_3. The line $\overline{\tau}$ parallel to τ_3 represents the average tangent drawn through (t_n,x_n) which then determines x_{n+1} according to Eq. (14).

Simple quadrature, where $f = f(t) \neq f(t,x)$ is easily performed by the Trapezoidal Rule obtained directly from Eqs. (11) and (12), without the need of any intermediate step:

$$x_N - x_0 = \int_{t_0}^{t_F} f(t)\,dt = \sum_{n=0}^{N-1} \int_{t_n}^{t_{n+1}} f(t)\,dt \doteq \tfrac{1}{2}\Delta t \sum_{n=0}^{N-1} [f(t_n)+f(t_{n+1})] \tag{15}$$

where $N\Delta t = t_F - t_0$, $t_n = t_0 + n\Delta t$.

Problem 3. Show that this approximation is equivalent to approximating the curve $f(t)$ by a line segment over each interval Δt, then finding the area under the polygon.

Horizontal Motion

Let us first examine the case of purely horizontal motion through the atmosphere with constant drag, for which an analytic solution to Newtonian resistance is quite simple. If $dx/dt = u>0$, then the equation of motion is

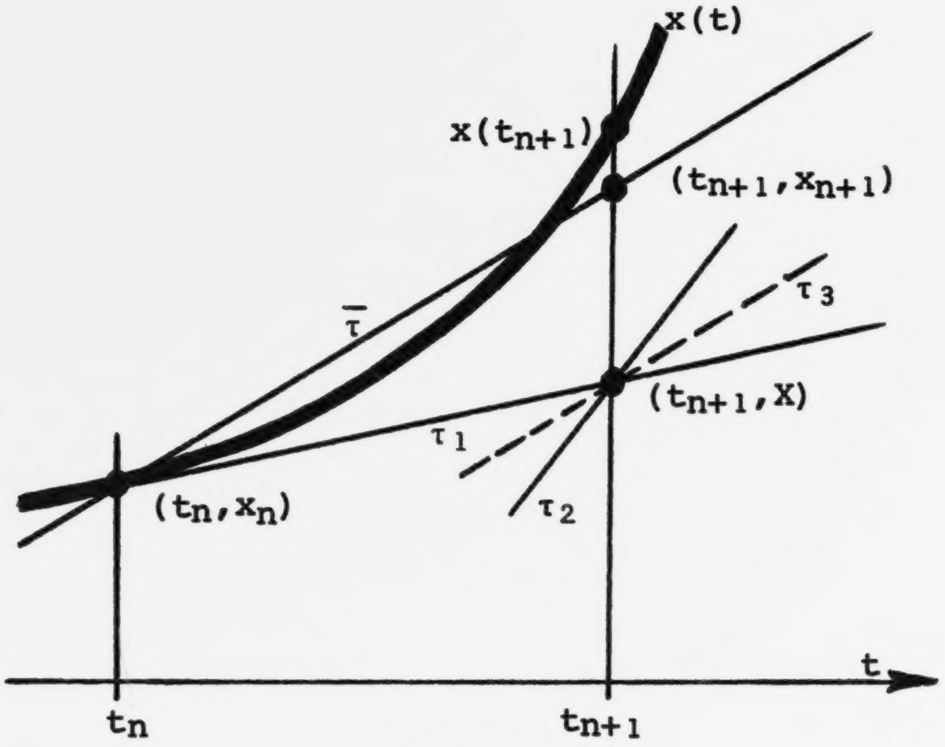


Figure 2. Second-order improved Euler method. The average slope over the interval is that of lines τ_3 and $\bar{\tau}$.

$$d^2x/dt^2 = du/dt = -(C_DA\rho/2M)u^2 \qquad (16)$$

which has the solution

$$u = u_0[(C_DA\rho/2M)u_0t + 1]^{-1} \qquad (17)$$

Integrating $dx/dt = u$ then yields

$$x = x_0+(2M/C_DA\rho)\ \ln[(C_DA\rho/2M)u_0t + 1] \qquad (18)$$

Comparing this solution for a Newtonian resistance to the Stokes' law solution, Eqs. (7), we note that in both cases the velocity gradually declines to zero; however, for large t Eq. (17) shows that $u \rightarrow 2M/C_DA\rho t$ independently of the initial speed! Furthermore, although Stokes' law resistance ultimately brings the body to rest at $x_\infty = x_0 + u_0/\alpha$, the Newtonian solution gives $x_\infty = \infty$. Both of these effects are due to the fact that the Newtonian resistance goes as u^2, very quickly cancelling the effect of a large initial velocity. Solving Eq. (11) for

$$t = (2M/C_DA\rho)[1/u-1/u_0] \qquad (19)$$

shows that the time required to attain some speed $u \ll u_0$ is practically independent of u_0 for u small enough. Furthermore, at low speeds the resistance rapidly decreases so that the body will not stop in a finite distance.

A realistic solution requires constant testing of the speed so that when N_R becomes less than 1,000, C_D is evaluated from some suitable function of N_R or from a table of values combined with an interpolation scheme. In air at one atmosphere pressure and about 15°C, $\rho = 1.2\ \mathrm{kgm/m^3}$, $\eta = 1.8\times10^{-5}\mathrm{n\text{-}sec/m^2}$, so a projectile of diameter 10cm will have $C_D = 0.5$ initially, if

$$L\rho u_0/\eta > 1{,}000 \quad \text{or} \quad u_0 > u_c = 1{,}000\eta/L\rho = 15\mathrm{cm/sec}$$

This approximation breaks down at a critical time (Eq. (19))

$$T_C = (2M/C_D A\rho)[1/u_c - 1/u_0] \tag{20}$$

when we must begin to account for changes in $C_D(N_R)$.

To see the effect that a variable drag coefficient has on the motion of a sphere let us perform a numerical integration of Eq. (16) and compare the results to the solutions of Eqs. (17) and (18). In performing numerical computations it is a good idea to "scale" the variables, i.e., to transform them to dimensionless quantities. This not only simplifies the programming and saves computing time, but can also afford insight into the physical quantities of interest. Expressing velocity in units of u_0 and time in units of $T = 2M/\rho A u_0$, Eq. (16) becomes

$$du/dt = -C_D(u)u^2 = f(u) \tag{21}$$

with initial conditions $t = 0$, $u = 1$. The computation of x from $dx/dt = u$ is quite straightforward once $u(t)$ is known, using the Trapezoidal Rule. In the above system of units, displacement x will be given in units of $L_0 = u_0 T = 2M/\rho A$. The basic algorithm is therefore

$$U = u_n - C_D(u_n)u_n^2\Delta t \ ; \quad u_{n+1} = u_n - \tfrac{1}{2}[C_D(u_n)u_n^2 + C_D(U)U^2]\Delta t \tag{22}$$

Figure 3 is the flow chart for the integration of Eq. (22); arrows indicate "assignment" statements--e.g., $t \leftarrow t + \Delta t$ means recall the value stored in the cell in the computer's memory which is addressed by the name "t", add Δt to it and store the new value in cell "t" in place of the old value of time.

The hexagonal box in the flow chart defining the function $f = -C_D u^2$ represents a separate "subprogram" for the evaluation of f (Figure 4); whenever f appears in the main program, the computer branches to this subprogram, evaluates f from the current value of u, and returns this value to the appropriate place in the main program. In this case, $f(u)$ is defined over three different regions:

(1) $1{,}000 \le N_R \le 10{,}000$, $C_D = 0.5$
(2) $0 < N_R \le 1$, $C_D = 24/N_R$
(3) $1 < N_R < 1{,}000$, C_D is found by linear interpolation on the table of logarithms given on page 178, in fair agreement with Figure 1.

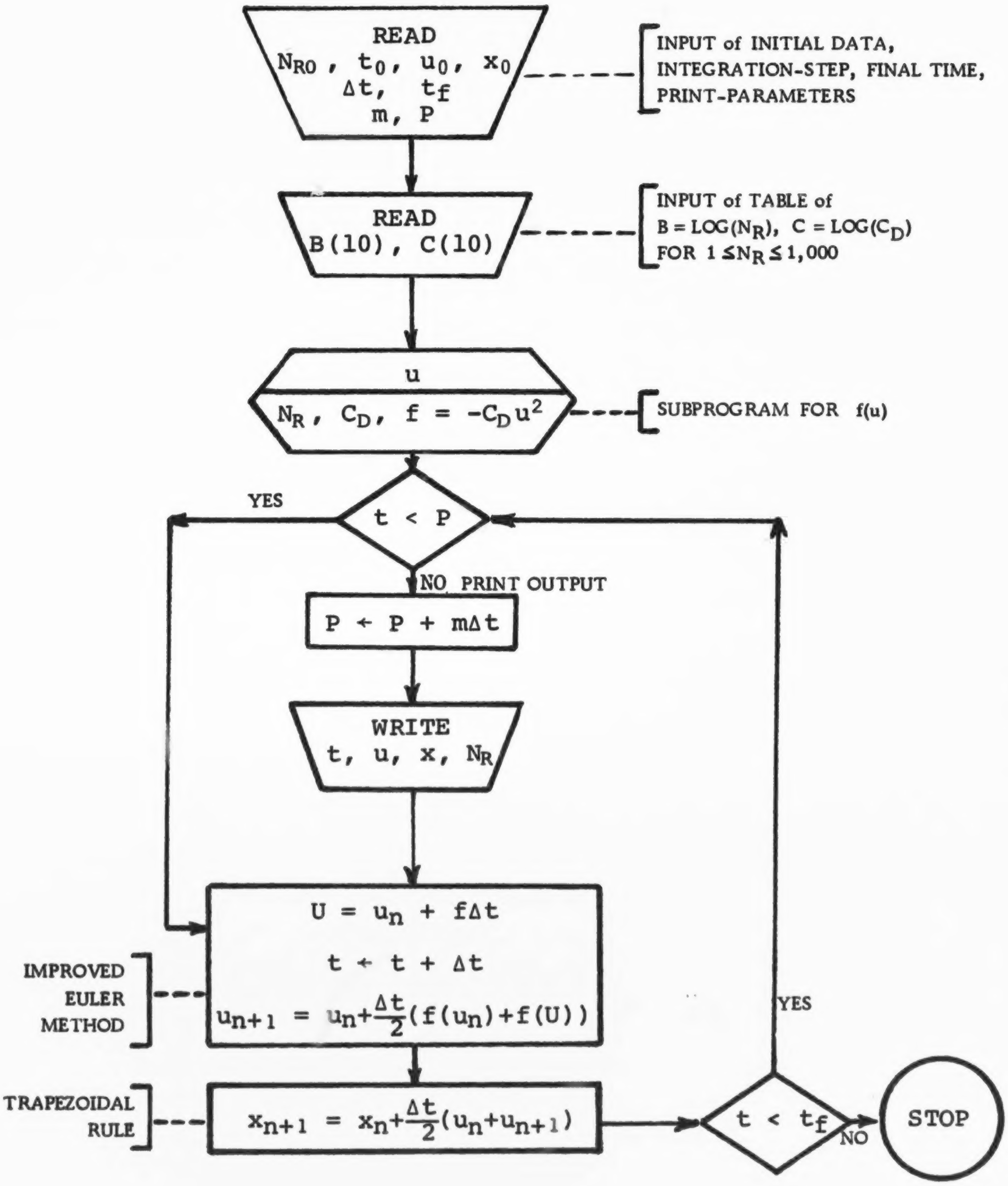

READ
N_R0 , t_0, u_0, x_0
Δt, t_f
m, P
INPUT of INITIAL DATA, INTEGRATION-STEP, FINAL TIME, PRINT-PARAMETERS
READ
B(10), C(10)
INPUT of TABLE of
B = LOG(N_R), C = LOG(C_D)
FOR 1 ≤ N_R ≤ 1,000
u
N_R , C_D, f = -C_D u²
SUBPROGRAM FOR f(u)
YES
t < P
NO PRINT OUTPUT
P ← P + mΔt
WRITE
t, u, x, N_R
U = u_n + fΔt
t ← t + Δt
u_n+1 = u_n + Δt/2 (f(u_n)+f(U))
IMPROVED EULER METHOD
TRAPEZOIDAL RULE
x_n+1 = x_n + Δt/2 (u_n+u_n+1)
t < t_f
YES
NO
STOP

Figure 3. Integration of $du/dt = -C_D(u)u^2$ by the improved Euler method.

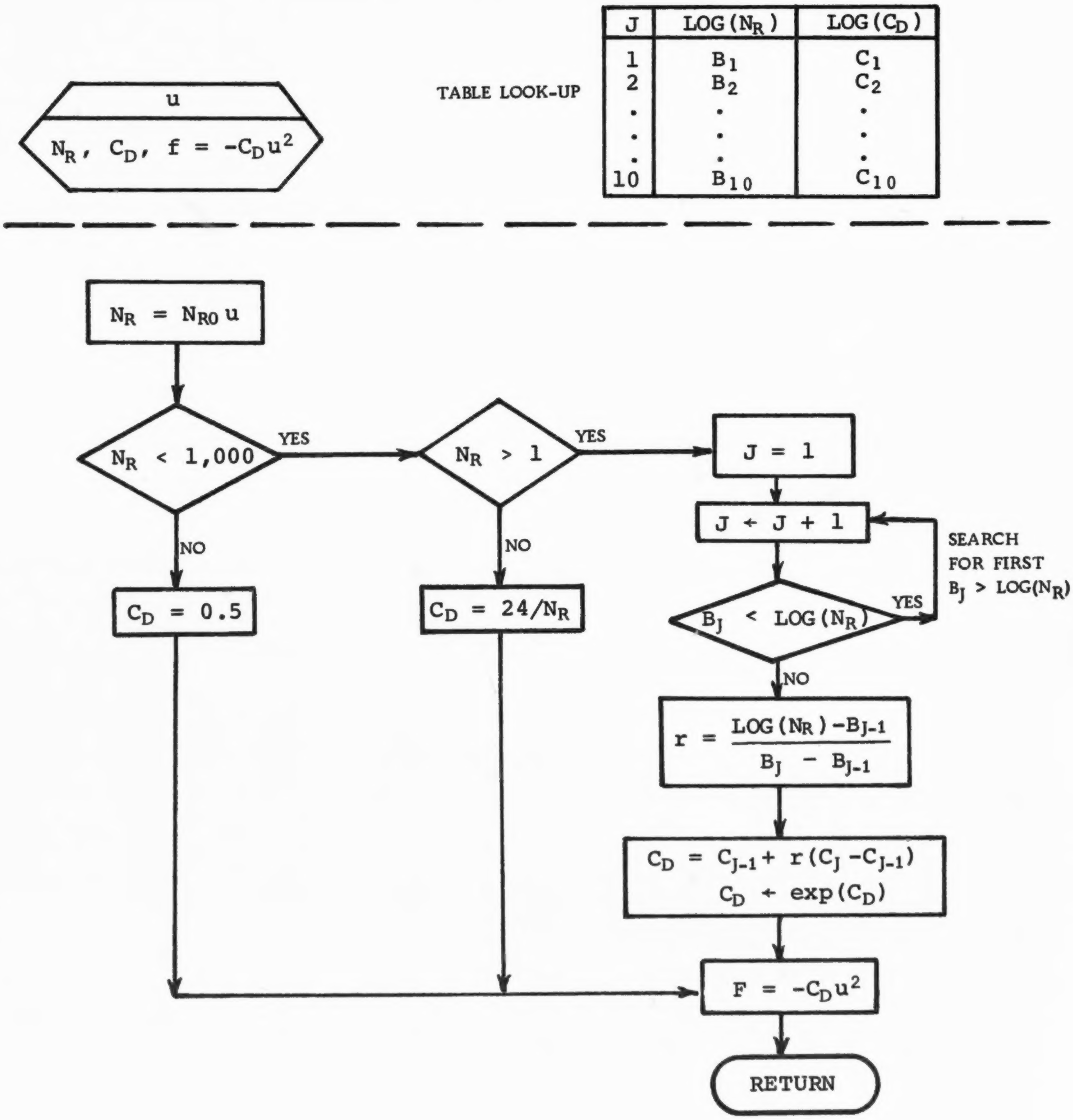

u
N_R, C_D, $f = -C_D u^2$
TABLE LOOK-UP
J
LOG(N_R)
LOG(C_D)
1
B_1
C_1
2
B_2
C_2
10
B_{10}
C_{10}
$N_R = N_{R0}u$
$N_R < 1{,}000$
YES
NO
$C_D = 0.5$
$N_R > 1$
YES
NO
$C_D = 24/N_R$
J = 1
J ← J + 1
SEARCH FOR FIRST $B_J >$ LOG(N_R)
$B_J <$ LOG(N_R)
YES
NO
$r = \frac{LOG(N_R) - B_{J-1}}{B_J - B_{J-1}}$
$C_D = C_{J-1} + r(C_J - C_{J-1})$
$C_D \leftarrow \exp(C_D)$
$F = -C_D u^2$
RETURN

Figure 4. Subprogram for evaluation of resistive force term, $f = -C_D u^2$.

N_R	$\log_e N_R$	C_D	$\log_e C_D$
1	0.00000	24.0	3.17805
2	0.69315	15.0	2.70805
5	1.60944	7.0	1.94591
10	2.30259	4.1	1.41099
20	2.99573	2.4	0.87547
50	3.91202	1.4	0.33647
100	4.60517	1.0	0.00000
200	5.29832	0.73	-0.31471
500	6.21461	0.57	-0.56212
1000	6.90776	0.50	-0.69315

Table 1. $C_D(N_R)$ for $1 < N_R < 1,000$.

The inclusion of the print parameters m and P is to avoid printing out the results at every integration-step, but only at "print-times" P occuring every m-th step at intervals of $m\Delta t$. This is purely for reasons of economy.

If the motion begins in the Newtonian regime, we can predict from Eq. (20) the time it takes before the transition to the Stokes regime begins; in dimensionless units,

$$t_c = \frac{T_c}{T} = \frac{\rho A u_0}{2M} \cdot \frac{2M}{C_D A \rho}\left[\frac{L\rho}{1,000\eta} - \frac{1}{u_0}\right]$$

$$\therefore \quad t_c = \frac{1}{C_D}\left[\frac{N_{R0}}{1,000} - 1\right] \qquad (23)$$

As a sample computation, for $N_{R0} = 10,000$ the improved Euler method algorithm, using integration-steps of $\Delta t = 0.1$, agrees with Eq. (11), the analytic solution, to within a maximum error of 0.03% over the range $0.0 \le t_c \le 18.0$, $10,000 \ge N_R \ge 1,000$, so that we need not be concerned about the accuracy of the method in this case; especially since, for smaller N_R, the accelerations are much smaller.

Continuing the computation into the Stokes regime shows the effect of the realistic increases in drag coefficient in slowing up the body, so that it comes to rest at approximately $x_\infty = x_0 + 9.73 L_0 T$ which is the area under the computed curve of Figure 5. As the graph indicates, percent differences in the computed and theoretical values of u are relatively unimportant until u has become quite small. To appreciate the importance of the realistic model, consider the difference in the predictions of x_∞. A Newtonian model predicts $x_\infty = \infty$, while a Stokes-law prediction yields

$$x_\infty = x_0 + v_0/\alpha = x_0 + Mv_0/6\pi\eta R = x_0 + (N_{R0}/12)(M/\rho\pi R^2)$$

or, in this case,

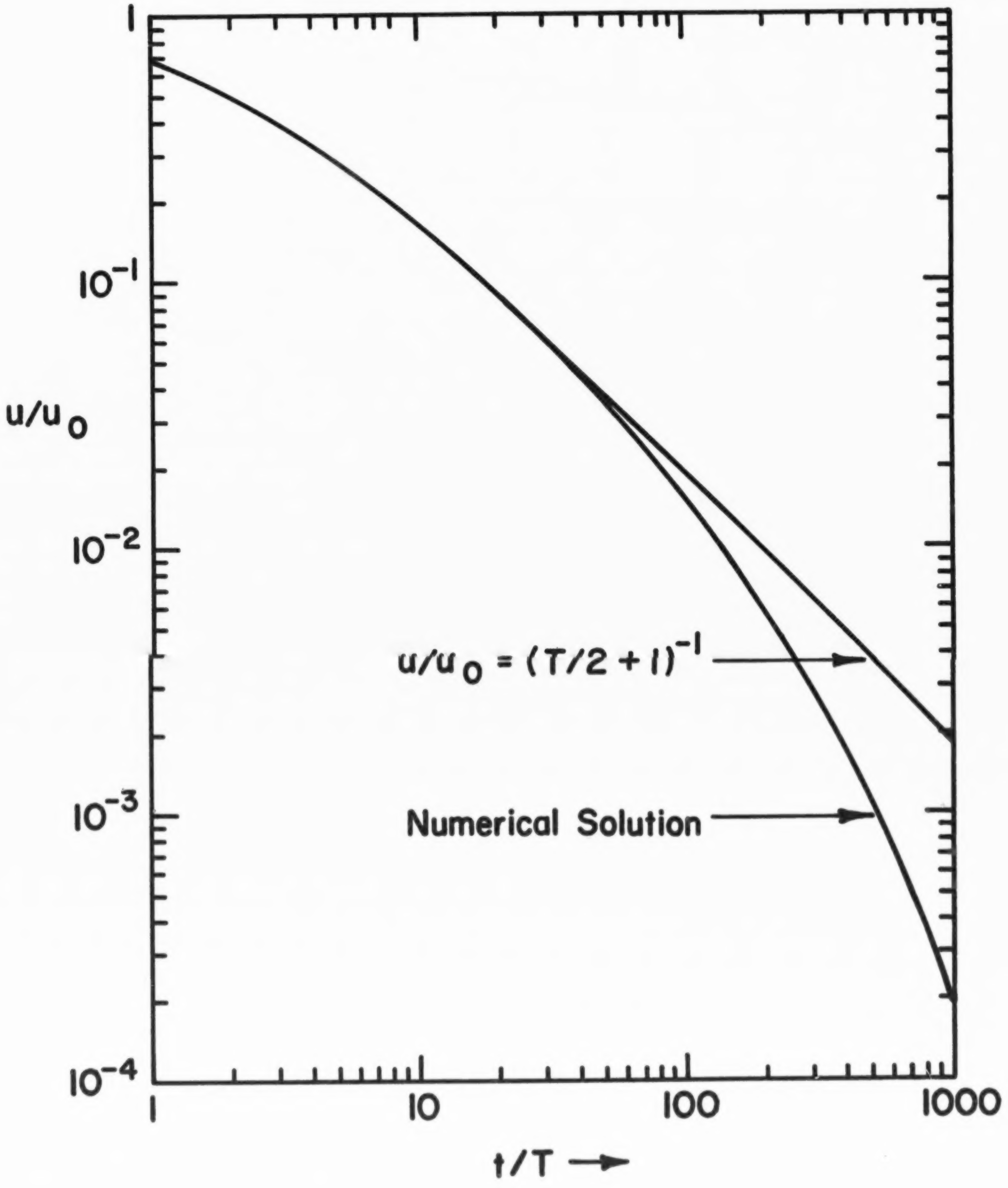


Figure 5. Comparison of computed solution and Newtonian solution for horizontal case.

$$x_\infty = x_0 + 833(M/\rho A) \qquad \text{(Stokes)} \qquad (24)$$

which is considerably greater than the numerical solution

$$x_\infty = x_0 + 19.5(M/\rho A) \qquad \text{(computed)} \qquad (25)$$

Note that $M/\rho A = L_0/2$ is the length of a cylinder swept out by M which contains a mass of fluid equal to M. The reason that both limiting solutions give ranges greatly in excess of the computed range is because (1) at low velocities the Newtonian drag becomes too small to stop the object, and (2) at large velocities $N_R >> 1$ Stokes' law gives a drag which is too small.

Problem 4. If ρ' is the density of the spherical object show that its stopping distance is proportional to the ratio ρ'/ρ and to its radius.

Problem 5. Construct the program for Figures 3 and 4 and compute $x(1,000)$ for $N_{R0} = 10,000$ (dimensionless variables, of course).

Problem 6. Show that in the limit of Stokes' law, $u \sim e^{-at}$ where $a = 24/N_{R0} = 24\times10^{-4}$.

Problem 7. How would you go about changing the integration-step in the middle of a computation?

Vertical Motion

Next we shall consider a less tractable case, that of vertical motion through the atmosphere. If we take y = altitude, positively increasing upward, then the equation of motion is

$$d^2y/dt^2 = dv/dt = -g-(C_D A\rho/2M)Vv \qquad (26)$$

It is similar in content to Eq. (16), but complicated by the appearance of gravitational acceleration, $-g$, and the fact that $\rho = \rho(y)$. For most problems of practical interest the variation in altitude is small compared to the Earth's radius, hence we will take g to be constant, $g = 9.8\text{m/sec}^2$.

Problem 8. If a twelve-pound iron cannonball is traveling at 30 miles per hour at sea level, compute the ratio $(2Mg)/(\rho v^2 C_D A)$.

To construct the algorithm for integrating second-order equations of the form

$$d^2y/dt^2 = f(t,y,dy/dt)$$

we must consider $v = dy/dt$ as a separate function and solve two simultaneous first order equations:

$$dy/dt = v \; ; \quad dv/dt = f(t,y,v) \qquad (27)$$

which we integrate numerically "in parallel"; computing the first-order predictions

$$Y = y_n + v_n\Delta t \ ; \quad V = v_n + f(t_n,y_n,v_n)\Delta t \tag{28}$$

and the average slopes

$$\bar{v} = (v_n + V)/2 \ ; \quad \bar{f} = [f(t_n,y_n,v_n) + f(t_{n+1},Y,V)]/2 \tag{29}$$

and, finally,

$$y_{n+1} = y_n + \bar{v}\Delta t \ ; \quad v_{n+1} = v_n + \bar{f}\Delta t \tag{30}$$

Before integrating, it will simplify matters to scale the variables as we did in the horizontal case, setting

$$v_0, \ T = 2M/\rho_S A v_0 \quad \text{and} \quad v_0 T = L_0 = 2M/\rho_S A$$

as the units of speed, time and space, respectively, where, for convenience, we take ρ_S = atmospheric density at sea level. Then the scaled equation of motion becomes

$$dv/dt = -G-C_D(\rho/\rho_S)Vv \ ; \quad G = gT/v_0 \tag{31}$$

Note that, since a particle may reverse its direction due to gravity, hence the sign of v may change, it becomes necessary to introduce the speed $V = |v|$ into the equation of motion, so that the resistive force is always opposed to the motion. This should be compared with Eq. (21).

Problem 9. Write Eq. (21) for a particle initially traveling to the left.

In constructing the algorithm we will need a new subroutine for the evaluation of ρ. In physics the "barometric equation," or the "law of atmospheres" is almost a commonplace. It states that for an ideal gas in which pressure and density are related by $P = \rho RT$, where R is a constant and T is the absolute temperature, the change in pressure with altitude of a column of gas is given by $dP = -\rho g dy$, just due to the weight of the gas in a slice of thickness dy. Substituting and integrating yields the barometric equation

$$P = P_S e^{-y/H} \ ; \quad \rho = \rho_S e^{-y/H} \tag{32}$$

where H is known as the scale height.

Problem 10. Derive Eq. (32) and evaluate the scale height at sea level, using the density given in Table 2 and one atmosphere of pressure.

Although this approximation is useful over $\Delta y \lesssim H$, it is not precisely correct since the temperature and composition, hence the scale height, vary with altitude as well, although much more slowly than ρ. Therefore, we shall compute ρ by linear interpolation on its logarithm as given in Table 2 (see page 182). At higher altitudes the atmosphere is so tenuous that its resistance may be

y (km)	$\log_{10} \rho$ (ρ in kgm/m^3)
0	.0882
10	.6226-1
20	.9491-2
30	.2653
40	.6097-3
50	.0098
60	.4832-4
70	.9465-5
80	.2874
90	.4937-6
100	.6794-7
110	.9723-8

Table 2

ignored. If in doubt about a particular case, try some order-of-magnitude estimates to satisfy yourself that this is true; if more data is needed it can be found in the Cospar International Reference Atmosphere (CIRA), 1961, North-Holland Publishing Company, Amsterdam (1961).

The flow chart is shown in Figure 6; it is a modification of Figure 3 to include a subprogram for evaluating ρ prior to evaluation of C_D since N_R depends on ρ. We shall assume the viscosity of the atmosphere to be constant, since it depends on the square root of the absolute temperature which varies by about 35% from 0 to 110km. The other assumption implicit in this model is that the radius of the object is much less than the scale height so we can be sure that the $C_D(N_R)$ dependence of Figure 1 holds.

Problem 11. Write a subprogram for the evaluation of the normalized density, ρ/ρ_S.

An ionosonde is a device launched into the ionosphere (y = 50-500km) to measure atmospheric properties, especially free electron densities and temperatures which strongly affect radio transmissions. Imagine an ionosonde in the form of a sphere one meter in diameter, mass 10kgm, launched from a rocket which maintains a steady acceleration of 3g during its powered vertical ascent. The power can be cut off by a signal from ground control, and sphere and booster caused to separate gently by means of air jets. Thus, we can consider the sphere separately after burn-out. The problem is this: if we wish to send the sphere to an altitude of exactly 100km, at what altitude do we shut off the rocket engines?

Let's begin by assuming that the rocket has reached some altitude y_0 and speed v_0 at burn-out. The standard kinematic formula for uniform acceleration gives $v_0^2 = 2(3g)y_0$, so that in a vacuum the ionosonde could ascend to altitude h

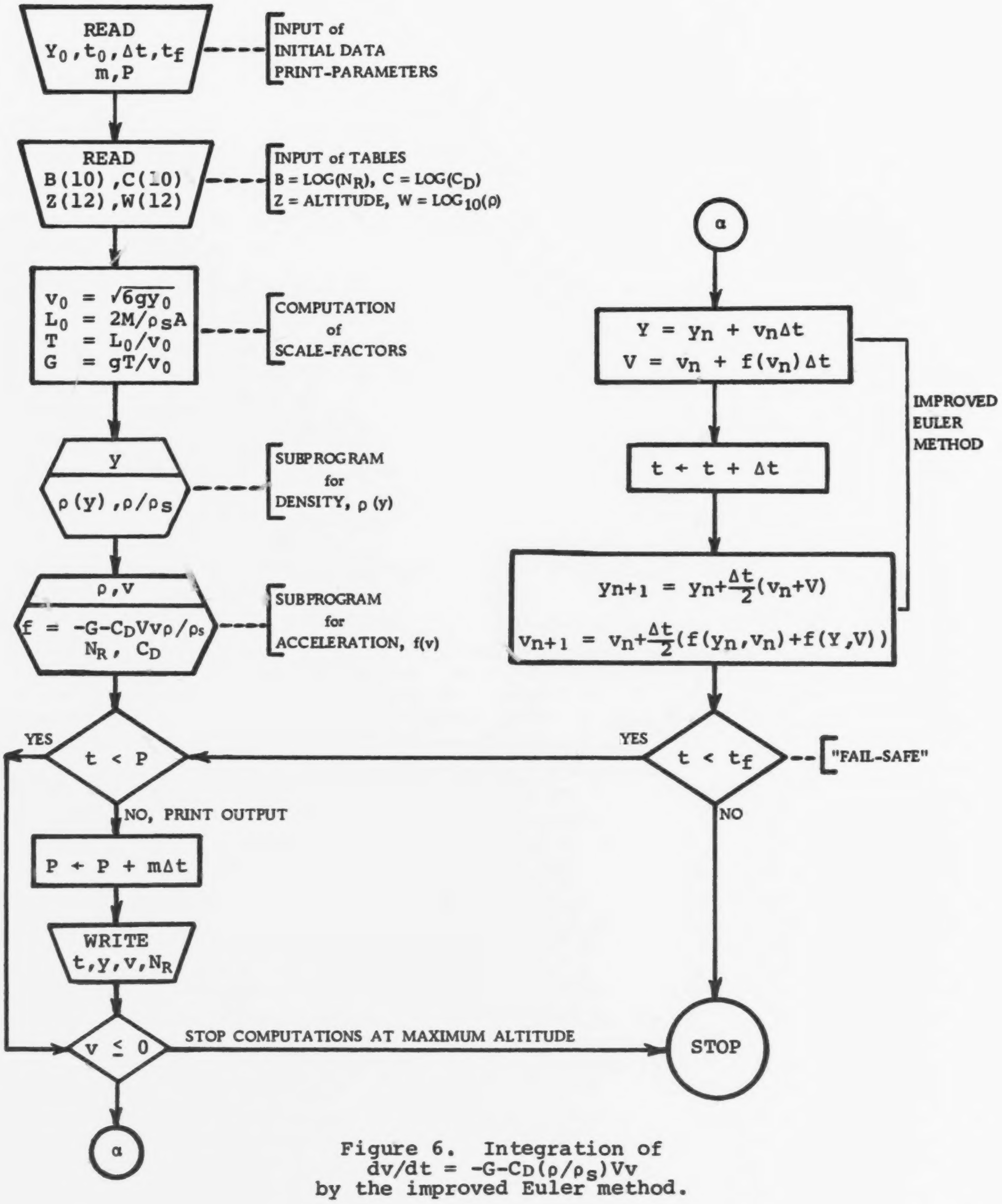

READ
$Y_0, t_0, \Delta t, t_f$
m, P
INPUT of
INITIAL DATA
PRINT-PARAMETERS
READ
B(10), C(10)
Z(12), W(12)
INPUT of TABLES
$B = LOG(N_R)$, $C = LOG(C_D)$
Z = ALTITUDE, $W = LOG_{10}(\rho)$
$v_0 = \sqrt{6gy_0}$
$L_0 = 2M/\rho_S A$
$T = L_0/v_0$
$G = gT/v_0$
COMPUTATION
of
SCALE-FACTORS
y
$\rho(y), \rho/\rho_S$
SUBPROGRAM
for
DENSITY, $\rho(y)$
ρ, v
$f = -G-C_D V v \rho/\rho_s$
N_R, C_D
SUBPROGRAM
for
ACCELERATION, f(v)
YES
t < P
NO, PRINT OUTPUT
$P \leftarrow P + m\Delta t$
WRITE
t, y, v, N_R
$v \leq 0$
STOP COMPUTATIONS AT MAXIMUM ALTITUDE
α
α
$Y = y_n + v_n \Delta t$
$V = v_n + f(v_n)\Delta t$
IMPROVED
EULER
METHOD
$t \leftarrow t + \Delta t$
$y_{n+1} = y_n + \frac{\Delta t}{2}(v_n + V)$
$v_{n+1} = v_n + \frac{\Delta t}{2}(f(y_n, v_n) + f(Y, V))$
YES
$t < t_f$
"FAIL-SAFE"
NO
STOP

Figure 6. Integration of
$dv/dt = -G-C_D(\rho/\rho_S)Vv$
by the improved Euler method.

$$h = y_0 + v_0^2/2g = 4y_0$$

before dropping back to Earth. For $h = 100\text{km}$, then $y_0 = 25\text{km}$, but due to frictional resistance, we would expect to have to burn fuel somewhat longer if we are to reach the desired altitude. Nonetheless, a first guess for $y_0 = 25\text{km}$ will do to start our computation. Note that at this altitude $\rho = 0.04\text{kgm/m}^3$, and the Reynolds number is approximately 2.7×10^6. Hence, the weight-to-drag ratio is $\rho_S G/\rho = Mg/(\frac{1}{2}C_D A\rho v_0^2) = 0.01$ at a minimum, implying that the resistive force will have some significant effect in determining y_0.

The following table gives the results of a typical trial-and-error computation:

y_0	h
25,000	28,600
30,000	38,070
40,000	82,290
42,000	98,150
42,200	99,700
42,260	100,130

Table 3. Maximum altitude, h, of ionosonde as a function of burn-out altitude, y_0, in meters. $R = 0.5\text{m}$, $M = 10\text{kgm}$. Improved Euler method, $t = 25$ units. $L_0 = 2M/\rho_S A = 20.7$ meters.

This represents a significant increase in our original prediction of 25,000 meters. To see the effect of increasing the mass without changing the cross-section of the sphere, we assume $M = 10\text{kgm}$, and repeat the computation. The result is the following:

y_0	h
25,000	54,640
30,000	87,840
31,500	97,460
31,900	100,010

Table 4. Maximum altitude, h, of ionosonde as a function of burn-out altitude, y_0, in meters. $R = 0.5\text{m}$, $M = 100\text{kgm}$. $\Delta t = 10$. $L_0 = 207$ meters.

Since your major fuel expense will probably depend mainly on the mass of the rocket and its burn-out altitude, whereas the pay-off depends on the amount of data you can collect, it is clearly advantageous to cram as much instrumentation as possible into a sphere of given size.

Problem 12. Compute burn-out altitude in the above example for some representative values of L_0 between 1 and 1,000; sketch a graph of $y_0(L_0)$.

Problem 13. Solve Eq. (31) for kinetic energy $w = v^2/2$ per unit mass of ascending body, as a function of altitude, $w = w(y)$. Assume $\rho = \rho(y)$ is any function of y.

Planar Motion: The Reentry Problem

We will close this discussion with the reentry problem, an example of a two-dimensional trajectory in a plane. In this case the equations of motion are a combination of Eqs. (5), (16) and (26). If we set $\underline{f} = -\frac{1}{2}C_DA\rho V(u\underline{i} + v\underline{j})$, $V = \sqrt{u^2+v^2}$ then the equations of motion are

$$d^2x/dt^2 = -(C_DA\rho/2M)Vdx/dt \tag{33}$$

$$d^2y/dt^2 = -g-(C_DA\rho/2M)Vdy/dt \tag{34}$$

Since the resistive term is a function of y, $u = dx/dt$ and $v = dy/dt$, we can omit discussion of $x(t)$ until after we have solved the problem for u,v, and y. Thus, we have to solve the three related dimensionless equations (using the same scale factors as in the vertical case):

$$du/dt = -C_D(\rho/\rho_S)Vu \tag{35}$$

$$dy/dt = v \tag{36}$$

$$dv/dt = -G-C_D(\rho/\rho_S)Vv \tag{37}$$

These equations are based on the assumption of a flat Earth; this is not unreasonable as long as the horizontal range of the trajectory is much less than the Earth's radius, $R_e = 6{,}400$km. A completed calculation should be tested for consistency to see if its horizontal range does meet this condition.

In "reentry" a rocket or satellite returns to the Earth's atmosphere from space or from Earth-orbit. Its altitude (assuming no lift from the atmosphere) is a monotonic decreasing function of time, and we may further simplify our statement of the problem by replacing time with altitude as the independent variable. Thus, we eliminate Eq. (36) and obtain

$$du/dt = (dy/dt)(du/dy) = vdu/dy \ ; \quad dv/dt = vdv/dy$$

and upon substitution into Eqs. (35) and (37) we obtain

$$du/dy = -C_D(\rho/\rho_S)Vu/v \ ; \quad dv/dy = -G/v-C_D(\rho/\rho_S)V \tag{38}$$

giving two independent first-order differential equations which can be integrated via computer with no more difficulty than the vertical case. We only need to (1) replace variable y by u and t by y, (2) change the output of the second subroutine from

$-C_D(\rho/\rho_S)V^2$ to $-C_D(\rho/\rho_S)V$, (3) alter the two equations for evaluation of the slopes du/dy and dv/dy, and (4) include a test for impact, $y < 0$.

Problem 14. Do it.

Problem 15. Formulate the equations of motion and initial conditions in polar coordinates for a round Earth.

As an example, assume that reentry begins at $y = 100$km, motion of the reentry body being little affected by the rarefied atmosphere above that altitude. A body returning to the Earth's gravitational field from infinity would, in a vacuum, acquire a speed of

$$V_0^2/2 = gR_e \qquad (39)$$

so we shall select $V_0 = 2gR_e \doteq 10$km/sec as a representative value. The second initial condition will be the assumption that the reentry angle $\theta_0 = 30°$. In calculations of this kind, it is customary to set up differential equations for V and $\theta = \tan^{-1}(-v/u)$ as indicated in Figure 7; however, we leave this as a problem.

Problem 16. Transform Eqs. (38) into equations in $V = \sqrt{u^2+v^2}$ and $\tan\theta = -v/u$.

To compute the range we can apply the Trapezoidal Rule to the computation of range

$$R = \int_0^t v\,dt = \int_{y_0}^0 (v/u)\,dy = \int_0^{y_0} \cot\theta\,dy \qquad (40)$$

similar to the way in which $x(t)$ was computed in Figure 3, the horizontal case. The tangent angle θ is also a useful quantity to compute, and it gives a vivid measure of the trajectory's departure from linearity.

The trajectory predicted by Eqs. (38) is the knee-shaped curve shown in Figure 7, the "knee" occurring in the region of maximum rate of change of θ, where friction has slowed the object sufficiently that gravity can pull it appreciably off course. The actual dimensions of the trajectory depends on $L_0 \sim M/A$, and if this ratio is large enough, the reentry body may hit the ground before reaching the knee.

In our example the flow is initially Newtonian, since at 100km

$$N_{R0} = \rho vL/\eta = (10^{.6794-7})(10^4)L/(2\times10^{-5}) \doteq 250L = 1{,}000$$

if we take $L = 4$m, a reasonable size. The results of the computations for a "ballistic coefficient" $\beta = Mg/C_DA = 5{,}000$n/m , a fairly typical figure, are shown in Table 5. (This quantity is often known as the "weight-to-drag ratio" although it is not dimensionless.)

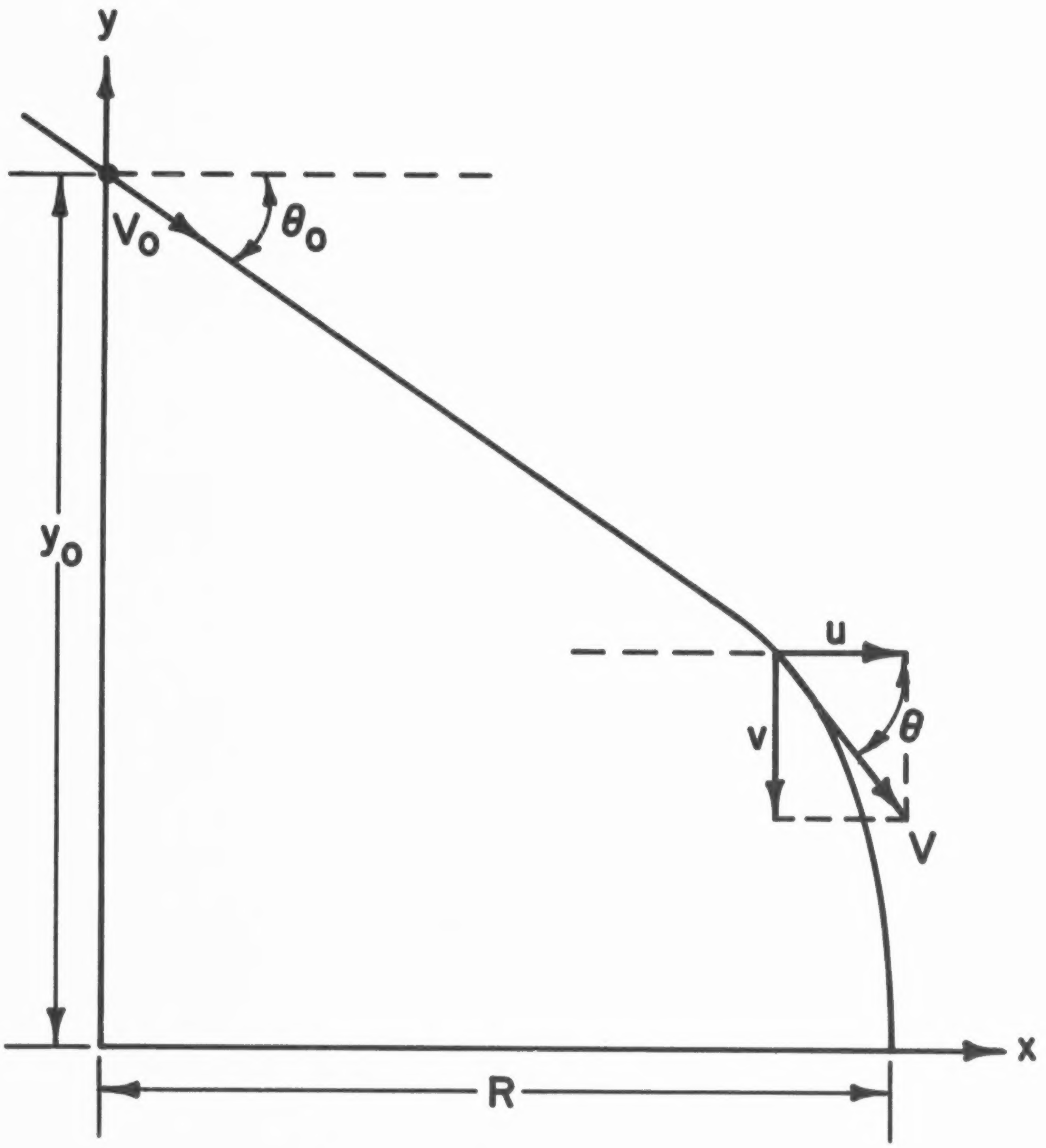

y
V_o
θ_o
y_o
u
v
θ
V
x
R

Figure 7. Reentry coordinates;
flat Earth approximation.

y (km)	x (km)	V (m/sec)	θ (°)
100.0	0.0	10,000.0	30.0
90.3	16.8	10,009.2	30.1
80.6	33.5	10,017.0	30.2
70.9	50.1	10,018.3	30.3
61.2	66.7	9,996.7	30.4
51.5	83.3	9,905.0	30.5
41.8	99.7	9,569.0	30.6
32.1	116.1	8,304.7	30.7
22.4	132.4	4,435.4	30.9
(continuing with an integration-step, $\Delta y = -2L_0$, $m = 4$)			
19.3	137.6	2,696.9	31.1
16.2	142.7	1 229.7	32.0
13.0	147.3	407.2	38.1
9.9	150.0	172.8	63.0
6.8	150.9	131.9	83.3
3.7	151.1	110.4	89.0
0.6	151.1	92.6	89.9

Table 5. Reentry trajectory for $V_0 = 10\text{km/sec}$, $\theta_0 = 30°$ at 100,000 m, ballistic coefficient $Mg/C_DA = 5{,}000\text{n/m}^2$, $L_0 = 388.2$ meters. Integration-step: $\Delta y = 5L_0$, print-step $m = 5$.

Problem 17. How would you add a computation of $t = t(y)$ to the program used to generate the results of Table 5 above?

TEACHER'S GUIDE

General Remarks

The material contained in the Student Manual represents an attempt to inject a certain amount of realism and vitality into a topic which is ordinarily treated in a stodgy and often incorrect manner in the texts, when it is treated at all. And yet its relevance in the Age of Speed will be quite apparent to the students. The concepts and techniques exhibited here can be applied to any motion through a fluid, which, with the exception of very slow motion or space travel, includes all cases of practical interest.

The progression from fairly simple notions to the more complex through a succession of higher-level problems is, hopefully, measured enough to be acceptable to junior- and senior-level mechanics students, and perhaps even to students of elementary physics who have had a thorough grounding in fundamental calculus. The material demands only an understanding of the meaning of first and second derivatives, integrals and the content of a differential equation. The rest is done by the computer. Some elementary experience with computers and their language and programming is a prerequisite, since this material is aimed explicitly at a more sophisticated student audience than the elementary level normally affords.

Those concepts which are to be swallowed whole as an act of faith are (1) the significance of the Reynolds number, (2) Stokes' law, and (3) the experimental data for $C_D(N_R)$ contained in Figure 1. If the student understands the meaning of an integral and a differential equation he should experience no difficulty in understanding the Trapezoidal Rule or the improved Euler method, although programming them may present some subtle difficulties, and the need for careful bookkeeping cannot be overestimated. If the second-order method causes severe difficulty, the student should be recommended to a first-order method which is the basis of the first approximation contained within the improved Euler method, i.e., $\Delta y \doteq \Delta x(dy/dx)$. It would also be an illuminating exercise to compare the accuracy (and the computation time) of the first- and second-order methods.

Since the direction of the resistive force is opposed to the velocity, $\underline{v}$, some pains have been taken to specify the vector nature of the problem (all vectors are underlined). Thus u and v are the x- and y-components of the velocity, while $V=|\underline{v}|=\sqrt{u^2+v^2}$ represents the speed. Figure 1, giving $C_D(N_R)$, was adapted from page 100 in Prandtl's book (see the bibliography at the end of this paper). One should point out that this approach (ignoring lift) can be used for any shape of projectile, so long as we have either a theory or data which gives C_D as a function of speed and the physical parameters of the problem. Students should easily see that the drag will depend, too, on the orientation or "angle of attack" of the projectile with respect to the fluid flow (moving coordinate system).

Numerical Integration and Quadrature

The improved Euler method discussed here is treated in detail in the book by McCalla (see bibliography) as a special case of Runge-Kutta methods; it is highly recommended for advanced students interested in this subject. Although the use of this method is not as general as that of the fourth-order, it affords a marked improvement in accuracy over the first-order method. Its use is recommended even for crude hand computations with a large integration-step as an easy way to obtain some idea of where a given differential equation is leading, before tackling it in earnest.

Figure 2 illustrates the relative accuracy of the first- and second-order methods over a rather large interval Δt. Although we have "derived" the Trapezoidal Rule as a special case of the improved Euler method, the argument should actually go the other way, as indicated by Eqs. (11) and (12). However, it could not be assumed that this result was familiar to the students, and the present approach was decided upon to emphasize the differential equations.

Horizontal Motion

It should be emphasized here that the analytic solution for the Newtonian flow must be self-consistent; i.e., it holds only until the Reynolds number drops below 1,000. The scaling of variables in Eq. (21) is not strictly necessary, but makes for more elegant statements as well as computational efficiency. If desired, values can be "unscaled" before output.

Scaling is theoretically desirable, also, because it ties in very naturally with notions of dimensional analysis and dynamic similarity. It also focuses attention upon which physical parameters, or combinations of them, are most important, and what physical meaning is to be ascribed to them. Thus, the Reynolds number is actually the ratio of the fluid's inertial reaction to being moved, $f_i = v\,dm/dt$, to the viscous force transmitted to the fluid, $f_v = A\eta\,dv/dr$, by the cylinder of area $A = \pi R^2$ swept out by the moving spherical object of radius R. The mass of fluid displaced per second is $dm/dt = \rho A v$ and $dv/dr \doteq v/2R$. Hence

$$f_i/f_v = (\rho A v^2)/(A\eta v/2R) = \rho v L/\eta = N_R \qquad (41)$$

It should be emphasized that this order-of-magnitude reasoning has an important place in physics, especially in fluid dynamics, space physics and astrophysics.

Note that the only physical input necessary for the scaled computation is the initial Reynolds number, N_{R0}, needed to determine the drag coefficients. Thus, situations having the same initial Reynolds number have the same dimensionless solution. However, the fact that the units of length and time may yet differ brings up the interesting possibility of "modeling." That is, if we double the mass-to-area ratio M/A, without changing the Reynolds number or initial speed, the units of length and time are both doubled. Thus, we can observe the phenomenon twice as long over twice the distance, and, after scaling the results to dimensionless quantities, apply them to the original case. By means of this sort of reasoning models are developed suitable for use in the laboratory which yield information about the real problem, which may be impossible to take indoors. In the simple example above, one might ask what would happen if we doubled the radius of a solid sphere? The mass would go up by a factor of eight, the Reynolds number would double and the units of length and time would change by a factor of 8/3--a totally different physical situation.

The following figure (Figure 8) gives a sample program written in BASIC for the computation of the horizontal drag problem illustrated in Figures 3 and 4. No particular attempt has been made to economize; the program has been made purposely redundant in some variables for the sake of clarity. For the same reason, and as a check on errors, the entire integration is performed numerically, although it would be more accurate to use the analytic formula, Eq. (17), for $N_R > 1{,}000$. Results are given in Table 6.

The program shown was written in elementary BASIC, in which all arrays are numbered from zero; to run in EXTENDED BASIC it is necessary to change 9 to 10 in statements 100, 180 and 190, and LET J = 1 in statement 310. Due to conversion of decimal to binary and the accumulation of round-off errors in P and T, it is necessary to insure against erratic behavior of the printing algorithm by setting $P = T-.00001$ in statement 400 and using final time $T1$ greater than the last time to actually be printed out.

If it is desired to change the integration-step Δt in the computation we change only the values of t_0, v_0, and x_0 to equal the last values computed before changing Δt. Print-parameters m and P may be changed if desired, but N_{R0} must remain constant; changing it is equivalent to a change in the physical parameters of the problem.

In the case of Problem 5, one can show that $u \sim e^{-24t/N_{R0}}$ (t dimensionless) by quoting the solutions of Eq. (7) for which $u \sim e^{-\alpha Tt}$ where

$$\alpha = 6\pi\eta R/M \ , \quad T = 2M/\rho A u_0 \ , \quad \alpha T = 24/N_{R0}$$

The dimensionless range is $x(1{,}000) = 9.687$.

```
100 DIM B(9),C(9)
110 READ NO,T,U,D,T1,M,X
120 REM NO=INITIAL REYNOLDS NUMBER
130 REM  T=INITIAL TIME
140 REM  U=INITIAL SPEED, X=INITIAL POSITION
150 REM  D=TIME-STEP
160 REM T1=FINAL TIME
170 REM  M=TIME-STEPS PER PRINT
180 MAT READ B(9)
190 MAT READ C(9)
200 REM  B=TABLE OF REYNOLDS NUMBERS, C=TABLE OF DRAG COEFFICIENTS
210 REM MORE ACCURATE TO DO LINEAR INTERPOLATION ON LOGARITHMS
220 GO TO 390
230 REM BEGIN TABLE LOOK-UP SUBPROGRAM
240 LET N=NO*U
250 IF N<1000 THEN 280
260 LET C1=0.5
270 GO TO 370
280 IF N>1 THEN 310
290 LET C1=24/N
300 GO TO 370
310 LET J=0
320 LET J=J+1
330 IF B(J)<LOG(N) THEN 320
340 LET R=(LOG(N)-B(J-1))/(B(J)-B(J-1))
350 LET C1=C(J-1)+R*(C(J)-C(J-1))
360 LET C1=EXP(C1)
370 LET F=-C1*U*U
380 RETURN
390 GOSUB 240
400 LET P=T-.00001
410 PRINT "  T                U               X                NR "
420 IF T<P THEN 460
430 LET P=P+M*D
440 PRINT T,U,X,N
460 LET U0=U
470 LET A=F
480 LET U=U+A*D
490 LET T=T+D
500 GOSUB 240
510 LET F=(A+F)/2
520 LET U=U0+F*D
530 LET X=X+.5*D*(U0+U)
540 GOSUB 240
550 IF T<T1 THEN 420
560 GO TO 610
570 DATA 10000,0,1,.1,10,10,0
580 DATA 0,.6932,1.6094,2.3026,2.9957,3.912,4.6052,5.2983
590 DATA 6.2146,6.9078,3.1781,2.7081,1.9459,1.411,.8755
600 DATA .3365,0,-.3147,-.5621,-.6932
610 END
```

Figure 8. BASIC program for horizontal drag problem, N_{R0} = 10,000.

```
          T                     U                   X                  NR
          0                     1                   0                  10000
          1.                    .666859             .811306            6668.59
          2.                    .500162             1.38693            5001.62
          3.                    .400124             1.8334             4001.24
          4.                    .333429             2.19817            3334.29
          5.                    .28579              2.50657            2857.9
          6.                    .25006              2.77371            2500.6
          7.                    .222272             3.00933            2222.72
          8.                    .200041             3.2201             2000.41
          9.                    .181853             3.41077            1818.53
          10.                   .166696             3.58482            1666.96
      570 DATA 10000,10,.166696,.2,20,10,3.58482
          10.                   .166696             3.58482            1666.96
          12.                   .14288              3.89318            1428.8
          14.                   .125018             4.16028            1250.18
          16.                   .111125             4.39588            1111.25
          18.                   .100012             4.60663            1000.12
          20.                   9.08427 E-2         4.79722            908.427
      570 DATA 10000,20.,9.08427 E-2,.5,101.,20,4.79722
          20                    9.08427 E-2         4.79722            908.427
          30                    6.13391 E-2         5.54077            613.391
          40                    4.55429 E-2         6.06819            455.429
          50                    3.57083 E-2         6.47101            357.083
          60                    2.90455 E-2         6.7928             290.455
          70                    2.42748 E-2         7.05817            242.748
          80                    2.07141 E-2         7.2823             207.141
          90                    1.79506 E-2         7.47511            179.506
          100                   1.57198 E-2         7.64308            157.198
      570 DATA 10000,100.,1.57198 E-2,5,501,10,7.64308
          100                   1.57198 E-2         7.64308            157.198
          150                   9.08233 E-3         8.24013            90.8233
          200                   5.93534 E-3         8.60724            59.3534
          250                   4.18058 E-3         8.85644            41.8058
          300                   3.08142 E-3         9.03604            30.8142
          350                   2.34995 E-3         9.17067            23.4995
          400                   1.84038 E-3         9.27473            18.4038
          450                   1.46314 E-3         9.35688            14.6314
          500                   1.17652 E-3         9.42256            11.7652
      570 DATA 10000,500.,1.17652 E-3,25,1501.,4.,9.42256
          500                   1.17652 E-3         9.42256            11.7652
          600                   7.84643 E-4         9.5191             7.84643
          700                   5.41475 E-4         9.58495            5.41475
          800                   3.83802 E-4         9.63031            3.83802
          900                   2.77248 E-4         9.66305            2.77248
          1000←                 2.03672 E-4         9.68689←           2.03672
          1100                  1.52777 E-4         9.70456            1.52777
          1200                  1.17412 E-4         9.71797            1.17412
          1300                  9.19882 E-5         9.72838            .919882
          1400                  7.23714 E-5         9.73657            .723714
          1500                  5.69379 E-5         9.743              .569379
```

Table 6. Solution to horizontal drag problem.
Note changes of integration-step Δt, print-parameter m.

Vertical Motion

In this case the acceleration is a function of the dependent variable y, as well as the speed v; therefore, it becomes necessary to consider $v = dy/dt$ as a separate independent variable related to y through a differential equation. Once the student overcomes his prejudice against first derivatives as being somehow inferior in status to the function itself (i.e., of lower degree), this should prove a very agreeable notion. This strategem is not only useful in numerical methods and the theory of differential equations, but is a precursor of the Hamiltonian formulation of mechanics which leads ultimately to quantum mechanics and the wave-particle dualism.

The variation of density $\rho = \rho(y)$ makes it necessary to include a second table look-up subroutine in the program. This subroutine must always be entered *before* evaluating $C_D(N_R)$ and $f(v)$, since the result depends on the density of the fluid. (See Figure 9.) Other questions which might be considered in connection with this problem are the computation of work done against friction during the powered phase of the ascent, the effects of diurnal fluctuations in atmospheric density as given in the 1965 edition of the COSPAR International Reference Atmosphere and the use of a rocket with constant thrust rather than constant acceleration. (The speed of a rocket with exhaust velocity V_0 starting from rest in vacuum is $v = V_0 \ln(M/M_0)$, M = mass of rocket.)

There is no reason why the subroutines for density and drag should be separate; they were kept distinct for the sake of clarity and ease of comparison with the preceding programs.

The solution to Problem 13 gives some idea of the complexity of an analytic approach to this problem. Substituting $w = v^2/2$ in Eq. (31), first setting $V = \pm v$ for the vertical ascent $(v > 0)$ or descent $(v < 0)$ yields

$$dw/dy = v(dv/dy) = dv/dt = -G \mp 2C_D(\rho/\rho_S)w \tag{42}$$

a first-order inhomogeneous equation in w which has the conventional solution

$$w = \exp\left[\mp\int_{y_0}^{y} Q(y')dy'\right]\left\{w_0 - \int_{y_0}^{y} G \exp\left[\pm\int_{y_0}^{y'} Q(y'')dy''\right]\right\} \tag{43}$$

$$Q(y) = 2C_D(\rho/\rho_S) \quad , \quad \rho = \rho(y)$$

If desired, one might also ask students whether or not $dw/dy=0$ at the maximum altitude of a vertical descent. The obvious tendency is to say "Of course, since the kinetic energy is a minimum there." The catch, however, is that although $w(y)$ has its minimum value over the allowed range of y, it is not an extremum for $y = h$, where $dw/dy = -G$. This point is obscured by the seeming complexity of the physical situation and the fact that $dw/dt = 0$ for $y = h$, since $w(t)$ *is* an extremum when $y(t) = h$. However, one can point

```
100 DIM B(9),C(9)
110 DIM Z(11),W(11)
120 REM Z=ALTITUDE IN TENS OF KMS; W=LOG(DENSITY) + 10.
130 READ Y,T,V,D,T1,M
140 REM NO=INITIAL REYNOLDS NUMBER
150 REM  T=INITIAL TIME
160 REM  V=INITIAL SPEED, Y8=REAL ALTITUDE IN METERS
170 REM  Y=NORMALIZED INCREMENT OF ALTITUDE, H=INITIAL ALTITUDE
180 REM  D=TIME-STEP
190 REM T1=FINAL TIME
200 REM  M=TIME-STEPS PER PRINT, H=INITIAL ALTITUDE(KM)
210 LET H=40000
220 MAT READ B(9)
230 MAT READ C(9)
240 REM  B=TABLE OF REYNOLDS NUMBERS, C=TABLE OF DRAG COEFFICIENTS
250 REM MORE ACCURATE TO DO LINEAR INTERPOLATION ON LOGARITHMS
260 MAT READ Z(11)
270 MAT READ W(11)
280 FOR I=0 TO 11
290 LET W(I)=W(I)-10.0882
300 LET Z(I)=10000*Z(I)
310 NEXT I
320 LET V9=SQR(6*9.8*H)
330 LET L9=80/(1.23*3.14159)
340 LET T9=L9/V9
350 LET G=9.8*T9/V9
360 LET P=T-.00001
365 GO TO 620
370 REM DENSITY, R8, AND DRAG, C1, SUBPROGRAMS
380 LET Y8=H+L9*Y
390 LET I=1
400 LET I=I+1
410 IF I=11 THEN 430
420 IF Z(I)<Y8 THEN 400
430 LET S=(Y8-Z(I-1))/(Z(I)-Z(I-1))
440 LET R8=W(I-1)+S*(W(I)-W(I-1))
450 LET R8=10↑R8
460 RETURN
470 LET N=N1*R8*ABS(V)
480 IF N<1000 THEN 510
490 LET C1=0.5
500 GO TO 600
510 IF N>1 THEN 540
520 LET C1=24/N
530 GO TO 600
540 LET J=1
550 LET J=J+1
560 IF B(J)<LOG(N) THEN 550
570 LET R=(LOG(N)-B(J-1))/B(J)-B(J-1))
580 LET C1=C(J-1)+R*(C(J)-C(J-1))
590 LET C1=EXP(C1)
600 LET F=-G-C1*R8*V*ABS(V)
610 RETURN
```

(continued)

```
620 GOSUB 380
630 LET N1=1.23*V9/1.8E-5
640 LET NO=R8*N1
650 PRINT "UNITS: ";V9;"M/SEC, ";L9;" M,  ";T9;" SEC "
660 PRINT
670 PRINT "T                   Y8                  V                   NR"
680 PRINT
690 GOSUB 470
700 IF T<P THEN 740
710 LET P=P+M*D
720 PRINT T,Y8,V,N
730 IF V<=0 THEN 960
740 LET Y0=Y
750 LET V0=V
760 LET A=F
770 LET Y=Y+V*D
780 LET V=V+A*D
790 LET T=T+D
800 GOSUB 380
810 GOSUB 470
820 LET F=(A+F)/2
830 LET V=(V0+V)/2
840 LET Y=Y0+V*D
850 LET V=V0+F*D
860 GOSUB 380
870 IF T<T1 THEN 690
880 GO TO 960
890 DATA 0,0,1,25,20000,16
900 DATA 0,.6932,1.6094,2.3026,2.9957,3.912,4.6052,5.2983
910 DATA 6.2146,6.9078,3.1781,2.7081,1.9459,1.411,.8755
920 DATA .3365,0,-.3147,-.5621,-.6932
930 DATA 0,1,2,3,4,5,6,7,8,9,10,11
940 DATA 10.0882,9.6226,8.9491,8.2653,7.6097,7.0098,6.4832
950 DATA 5.9465,5.2874,4.4937,3.6794,2.9723
960 END
```

Figure 9. BASIC program for vertical drag problem, y_0 = burn-out altitude, $V_0 = 6gy_0$, M = 10kgm, diameter = 1 meter.

```
UNITS:  1533.62 M/SEC,  20.7031  M,   1.34995E-2  SEC

  T                 Y8                V                NR
 0                 40000             1                348218.
 400               46607.4           .67599           94496.2
 800               51704.8           .567734          40394.4
 1200              56122.6           .503109          20950.7
 1600              60079.3           .454121          11702.2
 2000              63662.8           .412142          6820.47
 2400              66914.7           .37361           4136.75
 2800              69855.6           .336861          2593.31
 3200              72496.6           .301114          1558.94
 3600              74844.3           .265937          964.145
 4000              76902.            .231046          612.97
 4400              78671.4           .196308          398.158
 4800              80153.6           .16166           260.594
 5200              81349.1           .127071          164.634
 5600              82258.3           9.25167E-2       101.515
 6000              82881.4           .057986          56.7773
 6400              83218.7           2.34708E-2       21.6078
 6800              83270.2          -1.10354E-2       10.0643
```

Table 7. Sample output from program of Figure 9, y_0 = 40km, $\Delta t = 25$, $m = 16$.

out that because the resistive term vanishes with the square of the speed, the situation at turn-around, $y = h$, is no different from that of a solution in vacuum where $w = w_0 - Gy$ and $dw/dy = -G$, or, in classical terms, $\frac{1}{2}mv^2 = \frac{1}{2}mv_0^2 - mgy$ (with dimensions). They might be asked to sketch their impression of the curve $w(y)$ from 0 to h and back. The final kinetic energy on impact is $w_f < w_0$ due to resistive losses.

Planar Motion: The Reentry Problem

The program for the solution of this problem is readily adapted from that of Figure 9, with only those changes which are indicated in Figure 10. This computation differs from the vertical case chiefly in that range x is also computed by the Trapezoidal Rule (statement #870), the increment Δy is negative, and initial conditions are given in terms of speed, V9 = V_0, and angle, T8 = θ. Since V9 is a physical parameter of the motion, it is carefully distinguished from the dimensionless speed, V8 = V/V_0 to facilitate program interruptions for changes in the integration-step. Similarly, normalized input $Y = (y-y_0)/L_0$ is distinguished from initial altitude $H = y_0$.

In a vacuum the trajectory of a ballistic projectile will be a conic section with the Earth at one focus. If the Earth's center is taken to be the origin of a nonrotating set of coordinates (see

Figure 11), the problem remains two-dimensional and the equations of motion are

$$d^2r/dt^2 - r(d\phi/dt)^2 + gR_e^2/r^2 - (C_D A\rho/2M)Vdr/dt = 0 \qquad (44)$$

$$(1/r)[d(r^2 d\phi/dt)/dt] + (C_D A\rho/2M)Vr(d\phi/dt) = 0$$

$$V^2 = (dr/dt)^2 + r^2(d\phi/dt)^2 \ ; \quad \tan\theta = (dr/dt)/(rd\phi/dt)$$

which can be transformed to a single equation in $\sec^2\theta = z(r)$:

$$d^2z/dr^2 + (f\csc\theta - 2/r)dz/dr + (2/r - 2f\csc\theta)(z/r) = 0$$

$$f = C_D A\rho/M$$

```
130 READ Y,V8,T8,V9,D,M,X
150 REM H=INITIAL ALTITUDE, Y=INCREMENT IN ALTITUDE
160 REM U,V=X,Y VELOCITY COMPONENTS
170 REM D=ALTITUDE INCREMENT (NEGATIVE)
180 REM X=RANGE, ABSOLUTE, V8=SPEED (SCALED), V9=ABS. SPEED.
190 REM T8=THETA, DIRECTION OF MOTION.
210 LET H=100000
320 LET U=V8*COS(T8/57.2958)
330 LET V=-V8*SIN(T8/57.2958)
335 LET L9=1500/(1.23*3.14159)
360 LET P=Y+.00001
640 LET NO=R8*N1*V8
670 PRINT "Y           V8          Y8          X           T8          NR"
700 IF Y>P THEN 740
720 PRINT Y;V8;Y8;X;T8;NR
730 IF Y8<=0 THEN 960
745 LET U0=U
760 LET A1=F*U/V
765 LET A2=F-G/V
770 LET U=U+A1*D
775 LET V=V+A2*D
780 LET V8=SQR(U↑2+V↑2)
790 LET Y=Y+D
820 LET A1=(A1+F*U/V)/2
825 LET A2=(A2+F-G/V)/2
830 LET U=U0+A1*D
840 LET V=V0+A2*D
860 LET V8=SQR(U*U+V*V)
870 LET X=X+(U/V+U0/V0)*D*L9/2
880 GOSUB 380
```

Figure 10. Alterations to vertical drag program of Figure 9 for the case of reentry, y_0 = 100km, V_0 = 10km/sec, $\theta_0 = \arctan(-v_0/u_0) = 30°$, radius, R = 2m, M = 3,200kgm.

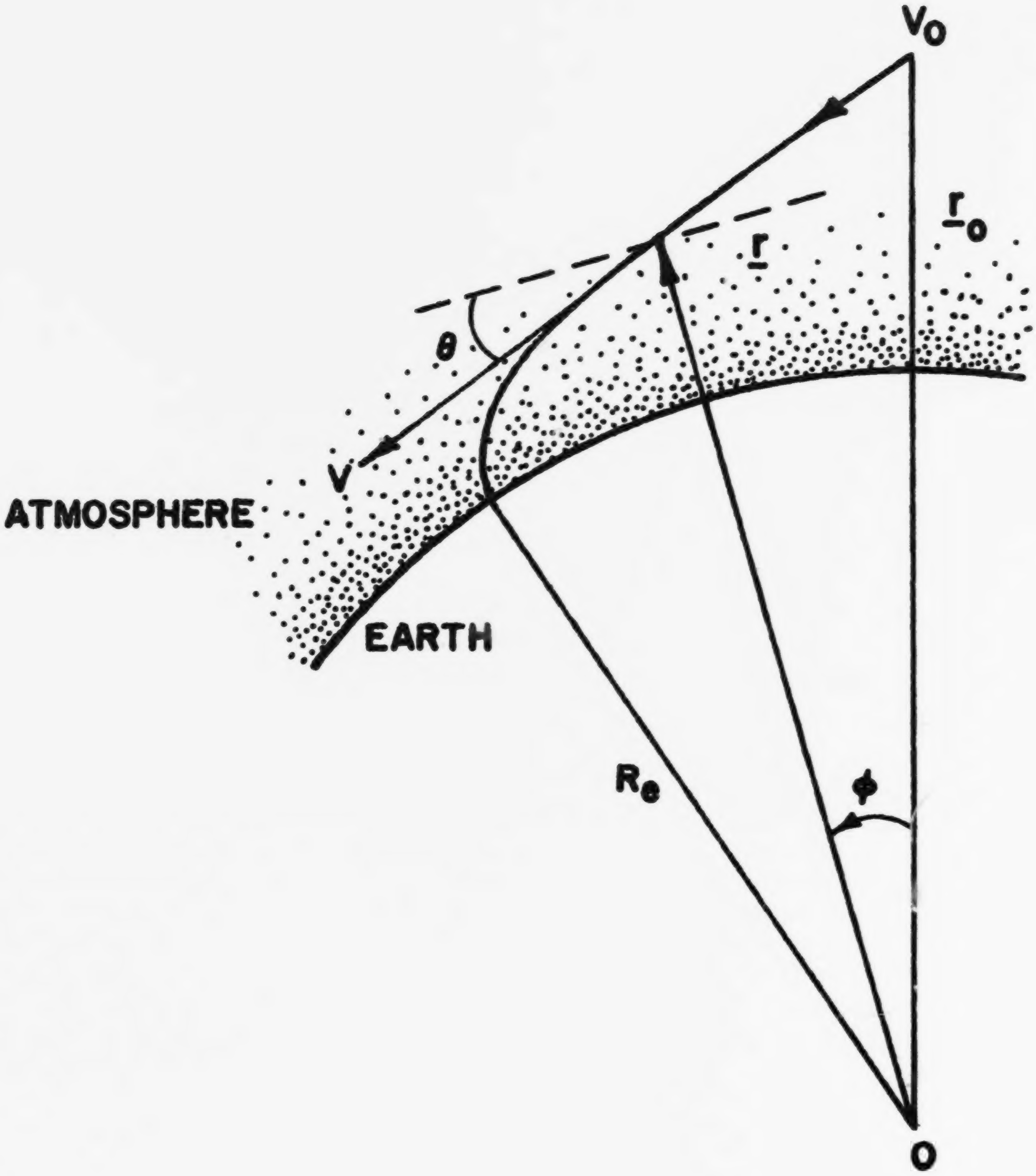

V_0
$\underline{r}_0$
$\underline{r}$
θ
V
ATMOSPHERE
EARTH
R_e
ϕ
0

Figure 11. The reentry problem in spherical (round-earth) coordinates. The geocenter is at 0.

To transform Eqs. (38) to the customary velocity coordinates V and θ we differentiate the expressions

$$V^2 = u^2 + v^2 \; ; \quad \tan\theta = -v/u \tag{45}$$

with respect to y to obtain

$$V(dV/dy) = -C_D(\rho/\rho_S)V^3/v - G \; ; \; \sec^2\theta(d\theta/dy) = G/uv \tag{46}$$

and substituting for $u = V\cos\theta$ and $v = -V\sin\theta$ yields, setting $w = \frac{1}{2}V^2$, the equations

$$dw/dy = -G + [2C_D(\rho/\rho_S)\csc\theta]w \tag{47}$$

$$d\theta/dy = -(G/2w)\cot\theta \tag{48}$$

Since $Q(y) = 2C(\rho/\rho_S)\csc\theta$ the resemblance of Eqs. (47) and (43) indicates that a closed implicit solution of the form $w = w[y,\theta(y)]$ and $\theta = \theta[y,\theta(y)]$ can be obtained. The solution for w is given by substitution for Q in Eq. (43), while the solution for θ is

$$\sec\theta = \sec\theta_0 \exp[-\int_{y_0}^{y}(G/2w)dy] \tag{49}$$

Barring special simplifying assumptions about ρ, it would be gilding the lily to say that these "solutions" are hopelessly intractable.

Bibliography

Blum, R., "Re-Entry Trajectories: Flat Earth Approximation," American Rocket Society Journal 32, 616 (1962).

Farina, M.V., Programming in BASIC, Prentice-Hall, Englewood Cliffs, N.J. (1968).

Martin, J.J., Atmospheric Reentry, Prentice-Hall, Englewood Cliffs, N.J. (1966).

McCalla, T.R., Introduction to Numerical Methods and FORTRAN Programming, Wiley and Sons, Inc., New York, N.Y. (1967) chap. 9.

Moe, M.M., "An Approximation to the Re-entry Trajectory," American Rocket Society Journal 30, 50 (1960).

Prandtl, L., and O.G. Tietjens, Applied Hydro- and Aeromechanics, Dover Publications, New York, N.Y. (1957).